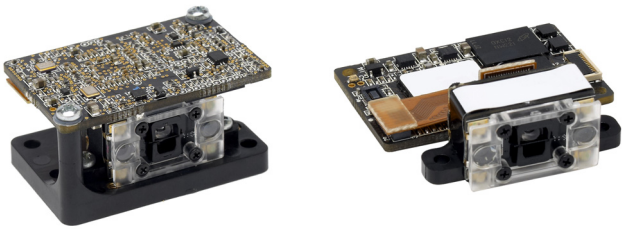
OMRON
MICROSCAN

MS-2D Engine User Manual



P/N 84-000008 Rev A

Omron Microscan Systems, Inc.
Tel: +1.425.226.5700 / 800.762.1149
Fax: +1.425.226.8250

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.

Technical Support

For technical support, e-mail: **helpdesk@microscan.com**.

Warranty

For current warranty information, see: **www.microscan.com/warranty**.

Omron 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

About the MS-2D Engine

The key features of the MS-2D Engine are:

- Ultra-compact size for embedded applications
- Dual field optics for both wide and high-density fields of view
- X-Mode technology for high-performance decoding
- USB 2.0 and RS-232 TTL interface options
- Low power draw (5V)

About This Manual

This manual provides complete information on setting up, integrating, and configuring the MS-2D Engine. The sections are presented in the order in which the MS-2D might be set up and made ready for operation.

Highlighting

Serial commands, highlighted command fields, and default command settings are highlighted in **rust bold**. Cross-references and web links are highlighted in **blue bold**. References to **ESP,** its toolbar headings (**Communications**, **Read Cycle**, **Symbologies**, etc.), menu topics, and other points of emphasis, are highlighted in **Bold Initial Caps**.

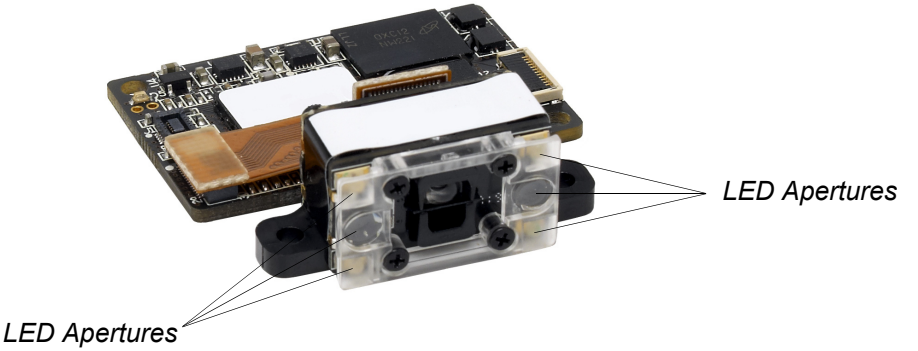
Host Communications

There are two ways to configure and test the MS-2D Engine:

- Omron Microscan's Windows-based **ESP** (Easy Setup Program) Software, which offers point-and-click ease of use and visual responses to user adjustments.
- Data Matrix programming symbols encoded with configuration commands.

Warning and Caution Summary

- Viewing the MS-2D's LED output with optical instruments such as magnifiers, eye loupes, or microscopes within a distance of 100 mm could cause serious eye injury.
- Maximum LED output: 141.2µW
- Location of the MS-2D's LED apertures:



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

IMPORTANT: The MS-2D is intended for connection to a UL-listed direct plug-in power unit marked Class II and rated 5 VDC at 3.5 Watts, or greater if using electrical accessories.

European models must use a similarly rated Class I or Class II power supply that is certified to comply with safety standard EN 60950.

Statement of Agency Compliance

The MS-2D conforms to the following Product Specifications:

EMC Emissions:	EN 55022: 2006 class B limits
LED Safety Compliance:	IEC 62471:2006 (Ed. 1.0)
ICES-003 Compliance:	ICES-003
FCC Compliance:	FCC Part 15, Subpart B

The product herewith complies with the requirements of the following Directive and carries the CE Marking accordingly:

- EMC Directive 2004/108/EC

This certification applies to the MS-2D as a stand-alone OEM engine and does not apply to the MS-2D as an integrated module. When integrating the MS-2D into another product, that product will still need to obtain any applicable CE certifications for the full product.

The MS-2D is RoHS compliant.

Statement of RoHS Compliance

All Omron Microscan readers with a ‘G’ suffix in the FIS number are RoHS-Compliant. All compliant readers were converted prior to March 1, 2007. All standard accessories in the Omron Microscan Product Pricing Catalog are RoHS-Compliant except 20-500013-01 and 98-000039-02. These products meet all the requirements of “Directive 2002/95/EC” European Parliament and the Council of the European Union for RoHS compliance. In accordance with the latest requirements, our RoHS-Compliant products and packaging do not contain intentionally added Deca-BDE, Perfluorooctanes (PFOS) or Perfluorooctanic Acid (PFOA) compounds above the maximum trace levels. To view the document stating these requirements, please visit:

http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:32002L0095:EN:HTML
and
http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2006:372:0032:0034:EN:PDF

Please contact your sales manager for a complete list of Omron Microscan’s RoHS-Compliant products.

This declaration is based upon information obtained from sources which Omron Microscan believes to be reliable, and from random sample testing; however, the information is provided without any representation of warranty, expressed or implied, regarding accuracy or correctness. Omron Microscan does not specifically run any analysis on our raw materials or end product to measure for these substances.
The information provided in this certification notice is correct to the best of Omron Microscan’s knowledge at the date of publication. This notice is not to be considered a warranty or quality specification. Users are responsible for determining the applicability of any RoHS legislation or regulations based on their individual use of the product.
In regards to “RoHS Directive 2011_65_EU” Omron Microscan produces Monitoring and Control Instruments as well as Industrial Monitoring & Control Instruments as defined within the directive. Omron Microscan has developed and is implementing a RoHS2 compliance plan with the intention of bringing all active products listed in our current marketing literature within full compliance as per the directive deadlines.
Key milestones for the transition plan are as follows:

- Complete internal product audit by July 2014.
- Initial “Monitoring and Control Instruments” RoHS2 compliant products available by December 2014
- Initial “Industrial Monitoring & Control Instruments” RoHS2 compliant products available by July 2015
- All new products introduced in 2015 are expected to be WEEE & RoHS2 compliant.

Omron Microscan will mark the products with the ‘CE’ marking that complies with the RoHS2 process to acquire ‘CE’ certification per the example given: Example >> Machinery directive + EMC directive + RoHS2 = Declaration of Conformity.

1 Demo Kit Quick Start

Contents

This section is designed to get your MS-2D Engine with development board up and running quickly. Following these steps will allow you to get a sense of the MS-2D's capabilities and to test symbol decode performance.

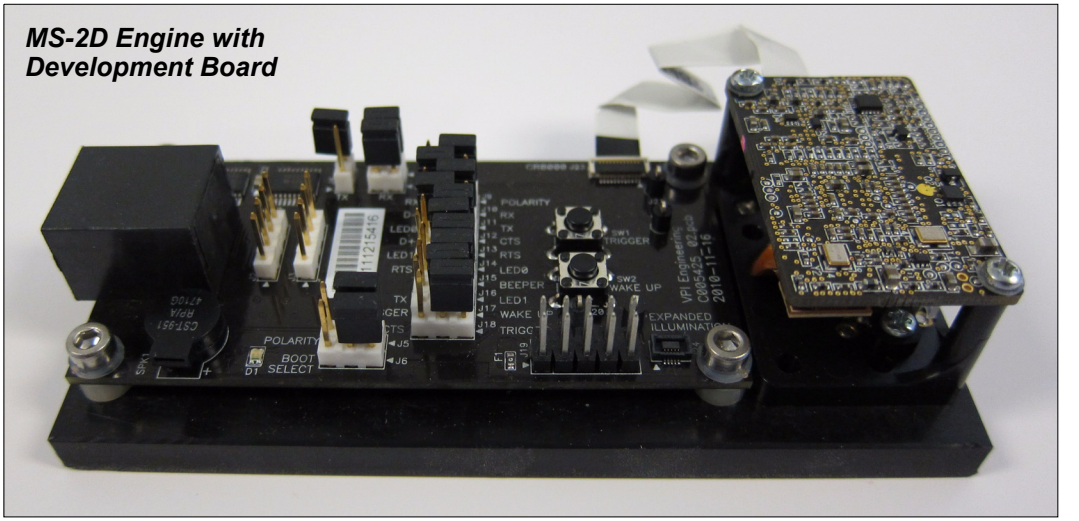
Additional setup information for installing the MS-2D into your application can be found in the subsequent sections.

Important: Although the development boards can be converted from USB to Serial and vice-versa with the appropriate RJ50 interface cable, the MS-2D Engine itself cannot be converted.

Step 1 — Check Hardware and Development Board

Caution: Be sure that all cables are connected **BEFORE** applying power to the system. Always power down **BEFORE** disconnecting any cables.

MS-2D Engine with Development Board



Item	Description	Part Number
1	Demo Kit, MS-2D, USB Engine, Decode, Bracket, Interface Board, 6 in. Ribbon	98-000224-01
2	Demo Kit, MS-2D, Serial Engine, Decode, Bracket, Interface Board, 6 in. Ribbon	98-000224-02
3	MS-2D Engine, USB, Tabs, 12 in. Ribbon, Standard Focus	FIS-MS2D-0001G
4	MS-2D Engine, USB, Bracket, 12 in. Ribbon, Standard Focus	FIS-MS2D-0002G
5	MS-2D Engine, Serial, Tabs, 12 in. Ribbon, Standard Focus	FIS-MS2D-0003G
6	MS-2D Engine, Serial, Bracket, 12 in. Ribbon, Standard Focus	FIS-MS2D-0004G
7	Ribbon Cable, MS-2D, 2 in.	61-000209-01
8	Ribbon Cable, MS-2D, 6 in.	61-000209-02
9	Ribbon Cable, MS-2D, 12 in.	61-000209-03

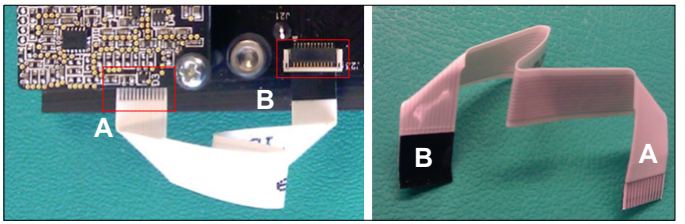
Remove the hardware from its packaging and inspect it for damage. Contact Omron Microscan if the product has been damaged during shipping.

The MS-2D is available in four versions:

- USB
- Serial
- With Mounting Tabs (Remote Decode Board)
- With Mounting Bracket (Attached Decode Board)

The MS-2D is also available as part of a demo kit, either USB or Serial. The USB demo kit is powered by USB and the Serial demo kit comes with an AC adapter and serial cable with power injector for the DC side of the power adapter.

Both versions of the demo kit include a z-folded ribbon cable to connect the decode board to the development board or host application. This cable is specially designed for the development board and is different from the 12” ribbon cables that ship with the MS-2D, as well as the cables shown in rows 7, 8, and 9 of the table above.



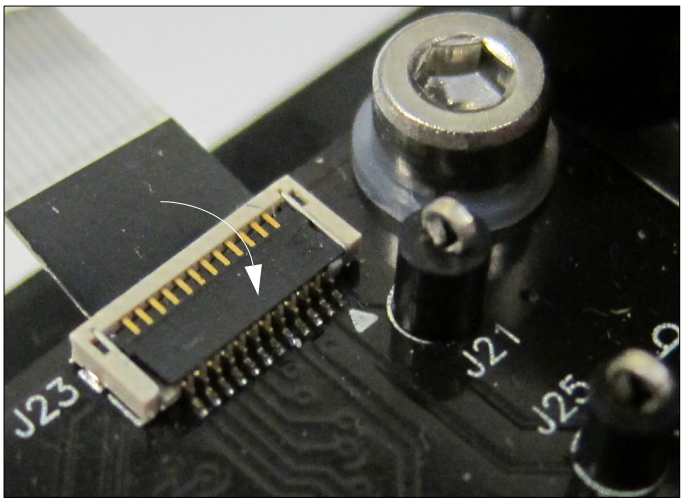
The contacts of either end of the development board interface cable must face the circuit board as shown at points **A** and **B**.

See **Development Board Connections** for details about the interface board that is provided as part of the MS-2D demo kit.

Attaching the Development Board Interface Cable

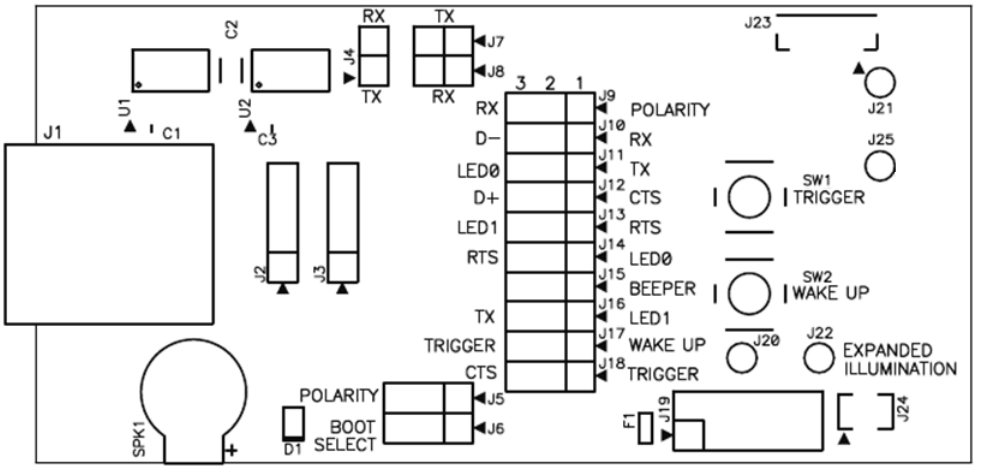
To attach the z-folded development board interface cable to the development board, first slide the end of the ribbon cable with leads facing the circuit board (downward) into the housing. Then carefully push the hinged clamp mechanism downward so that it covers the contact points, as shown below.

Caution: The connector housing and clamp mechanism are fragile.



Important: The same procedure described above must be used to attach the other end of the development board interface cable to the underside of the decode board. The only difference is that instead of facing down, the leads must be facing up.

Development Board Connections



Demo Kit Overview

The MS-2D demo kit includes everything needed to integrate the MS-2D scan engine into a target design. Omron Microscan provides a complete scan engine, development breakout board, and all documentation required to evaluate and integrate the MS-2D quickly.

Scan Engine

The demo kit comes with a MS-2D Engine and bracket, which includes the imager and decode board integrated into a single assembly.

Development Board

The development board is the main user interface to the MS-2D demo kit. It provides access to all features of the MS-2D, including debug and development resources.

Interface

J23 and J24 connect to the MS-2D decode board. J23 connects to the host interface and expanded illumination connector respectively.

Trigger/Wake Up Switches

SW1 and SW2 allow the user to trigger a decode and wake the unit up from its sleep state, respectively. If the unit is in a sleep state when the trigger is pushed, the MS-2D will automatically wake up before performing a decode.

Scan Interface

J1 interfaces to an RJ50 connector, which carries both USB and RS-232 signals to an external interface. The connector also provides a trigger signal to activate the MS-2D remotely.

Indicators

The development board includes a speaker (SPK1) for audible indication as well as a bi-color LED (D1) for visual indication.

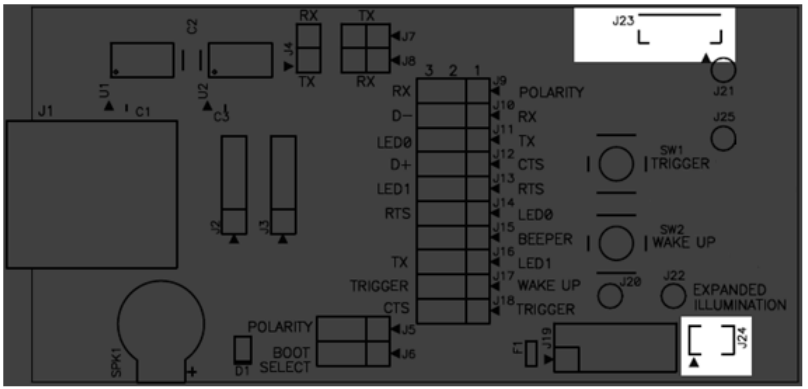
Configuration Jumpers

A group of jumpers allow the development board to reconfigure and access different features of the MS-2D. J7 and J8 configure auxiliary serial port features that appear on J2 and J3. J5 configures serial port polarity, and J6 in conjunction with SW2 controls which boot mode the MS-2D enters upon powering up. Finally, the MS-2D host port configuration can be changed via the jumper block J9-J18.

Auxiliary Headers

J2 and J3 provide auxiliary and debug serial communications to the MS-2D, and J19 provides access to all of the expanded illumination I/O.

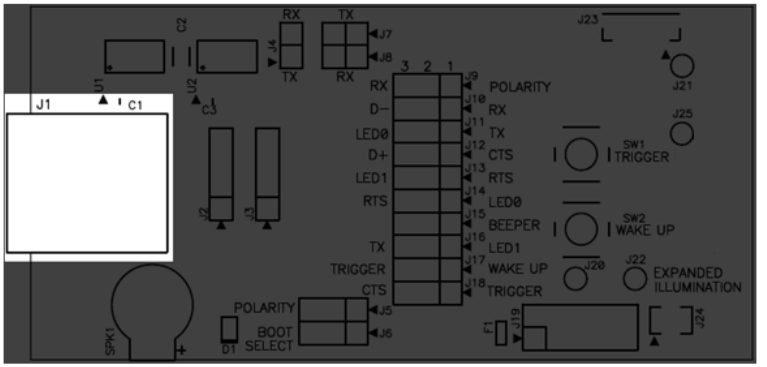
MS-2D Connections



The MS-2D connects to the development board via J23 and J24. (J24 allows you to connect the development board to the engine for external illumination. J24 is also connected to J19, the **Illumination Expansion Header**.) Contact Omron Microscan for the flex cable to connect J24 on the development to the MS-2D's decode board.

See **Host Interface Pinouts (RS-232)** and **Host Interface Pinouts (USB)** for the RS-232 and USB pin assignments of J23.

RJ50 System Header

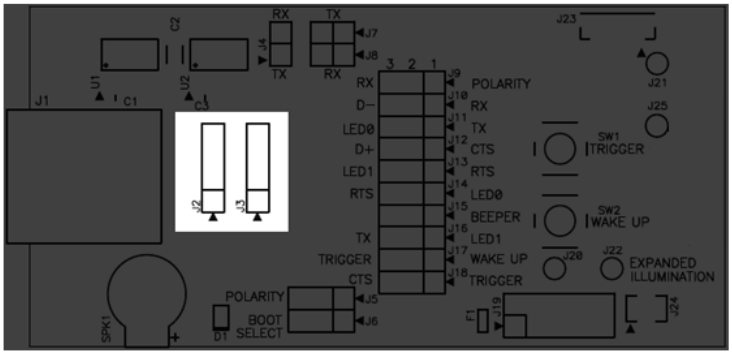


The majority of system communications go through the RJ50 System Header. The RJ50 cable has RS-232, USB, and trigger connections, and provides power to the demo kit.

RJ50 System Header Pin Assignments

Pin	Description
1	VIN
2	USB DATA-
3	USB DATA+
4	RS-232 RX
5	RS-232 CTS
6	RS-232 TX
7	RS-232 RTS
8	nTrigger
9	N/C
10	GND

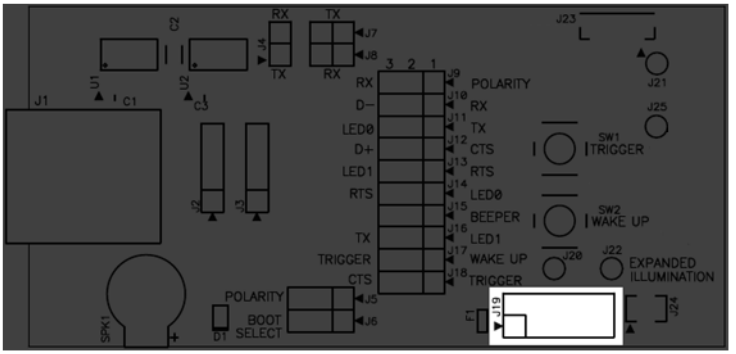
Auxiliary and Debug Serial Headers



J2 and J3 provide auxiliary communications to the MS-2D for development and test purposes. J2 is a serial console interface to the Linux kernel operating on the MS-2D engine, and J3 is attached to RS-232 Port 4.

Illumination Expansion Header

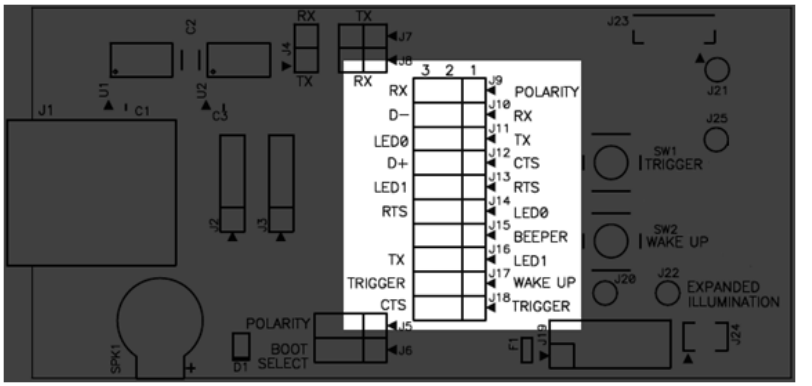
Contact Omron Microscan for additional information about the Illumination Expansion Header.



Pin	Description
1	VIN
2	VIN
3	Illumination I/O 0/RS-232 Port 4 TX
4	Illumination I/O 1/RS-232 Port 4 RX
5	Illumination PWM 1
6	Illumination I/O 2
7	Illumination I/O 3
8	Illumination I/O 4
9	GND
10	GND

Development Board Jumpers

Mode Configuration

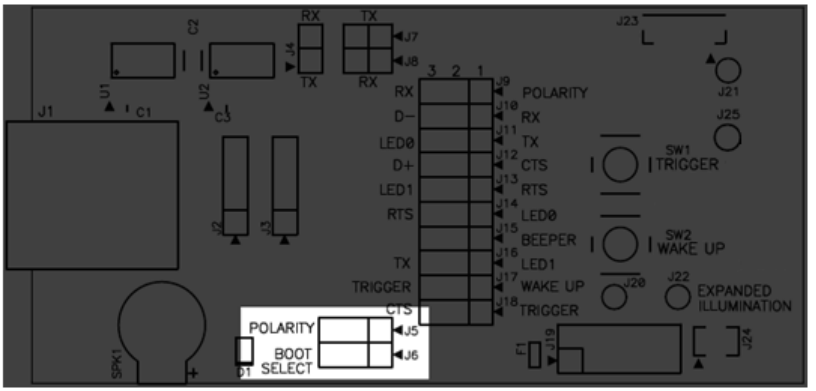


The jumper block of J9-J18 configures the signals between the MS-2D and RJ50 connector. This is done by shorting pins 1-2 or 2-3 on each jumper. Refer to the table below for configuring these pins:

Jumper	RS-232 Kit	USB Kit
J9	1-2	1-2
J10	1-2	2-3
J11	1-2	1-2
J12	1-2	2-3
J13	1-2	1-2
J14	1-2	1-2
J15	1-2	1-2
J16	1-2	1-2
J17	1-2	1-2
J18	1-2	1-2

Important: Although the development board can be converted from USB to Serial and vice-versa with the appropriate RJ50 interface cable, the MS-2D Engine itself cannot be converted.

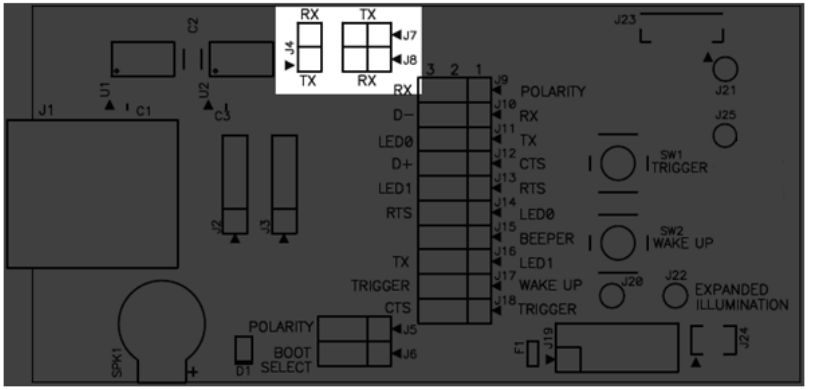
Serial Polarity and Boot Select Jumpers



J5 selects whether or not the primary RS-232 data are inverted, and the combination of J6 and SW2 will select the boot mode the kit comes up in. The RS-232 data will be inverted if J5 has pins 2-3 bridged, and will remain non-inverted if the jumper is left in the default position of 1-2. Refer to the table below for boot mode selection:

Boot Mode	J6	SW2	Description
1	1-2	Open	Normal Mode – Reader Application
2	1-2	Depressed	Upgrade Mode – Linux Application to upgrade Reader Application
3	2-3	Open	Factory Restoration Mode
4	2-3	Depressed	U-Boot Upgrade mode – Upgrade Linux OS and Filesystem

Serial Debug Jumpers



The serial debug jumper block of J4, J7, and J8 determine whether or not RS-232 Port 4 is routed to J3, and allows the user to tap off of the debug port on J2. J7 and J8 are shorted to connect MS-2D RS-232 Port 4 to J3.

Step 2 — Install ESP

ESP Software can be found in the **Download Center** at **www.microscan.com**.

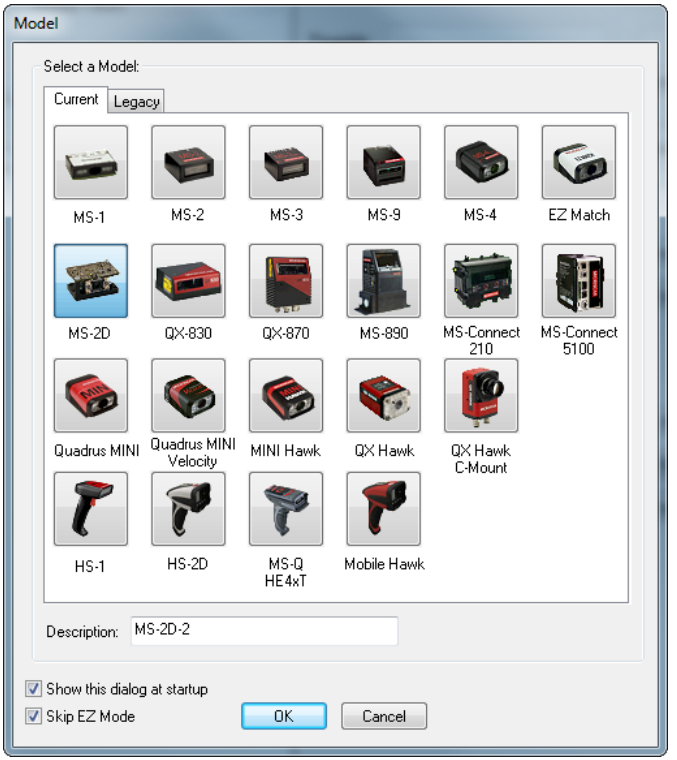
1. Click “Setup.exe” and follow the prompts in the installation wizard.
2. Click on the ESP icon to run the program.

Minimum System Requirements

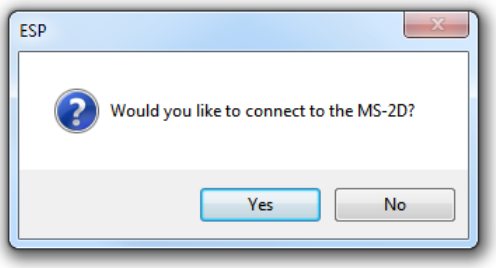
- 233 MHz Pentium PC
- Windows 8, 7, Vista, or XP operating system (32-bit or 64-bit)
- Internet Explorer 6.0 or higher
- 128 MB RAM or greater
- 160 MB free disk space
- 800 x 600 256 color display (1024 x 768 32-bit color recommended)

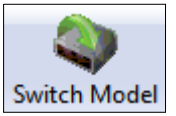
Step 3 — Select Model

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



1. Click the button showing the MS-2D Engine.
2. Click **OK**.
   Note: You can also double-click the MS-2D Engine button to make your selection.
3. Click **Yes** when this dialog appears:





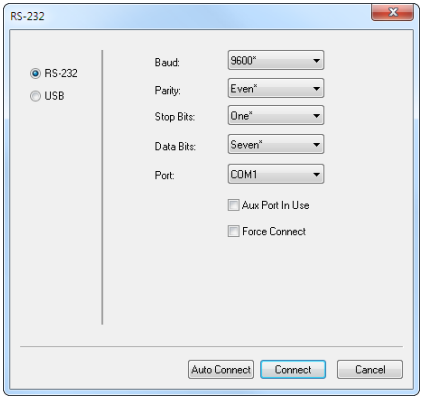
Note: If you need to select another model later, click the **Switch Model** button near the top of the screen or use **Model > New Model** in the menu toolbar.

Step 4 — Connect

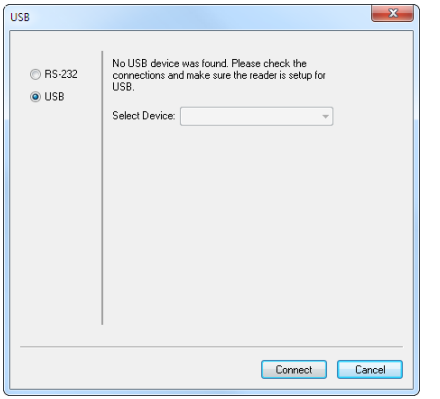
Connection Wizard

To connect using the Connection Wizard:

- Click **Connect** on the menu toolbar, and then select **Connection Wizard**.
- Select **RS-232** or **USB** to activate the appropriate display.
- Configure settings as required by the application, and click **Connect**.



RS-232 Connection Wizard

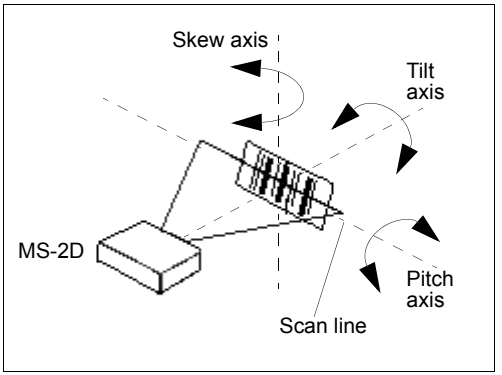


USB Connection Wizard

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

Step 5 — Position the MS-2D Engine

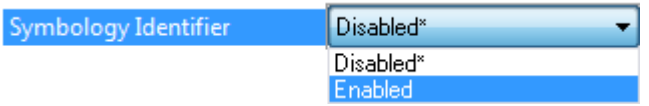
- Set up a symbol approximately 4” (10 cm) from the front of the 2D Engine.
- Avoid bright light or infrared light from other sources, including other readers.
- Pitch the MS-2D or symbol at a minimum of **±15°** to avoid specular reflection (the return of direct, non-diffused light).
- Avoid excessive skew or pitch. Maximum skew is **±30°**; maximum pitch is **±30°**.
- Center the blue bars on the center of the symbol and press the LED button to scan.



MS-2D and Symbol Orientation

Note: Code 39 is the default symbology enabled. If you are uncertain of your symbology, perform the following steps:

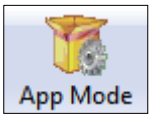
1. Enable all symbologies using **ESP**.
2. Enable **Symbology Identifier** at the bottom of ESP’s **Symbologies** tree control.



3. Decode the symbol and compare the symbology identifier character to the list on page **6-19** to determine your symbology.
4. Disable all other symbologies.

Step 6 — Configure the MS-2D Engine in ESP

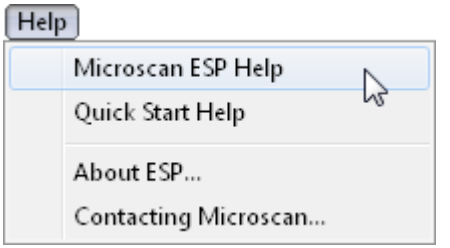
To make setup changes, click the **App Mode** button.



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

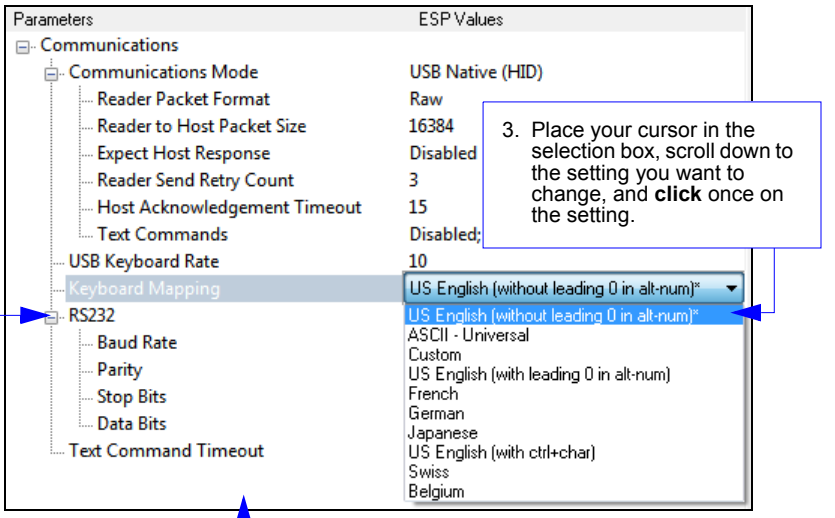
- Click the **EZ Mode** button to return to EZ Mode.
- Click the **Autoconnect** button to establish communication.
- Click the **Send/Recv** button to send or receive commands.
- Click the **Switch Model** button to open the model menu, or to return to a previous model.
- Click the **Parameters** button to show the tabbed tree controls for Communication, Read Cycle, and Symbologies.
- Click the **Terminal** button to display decoded symbol data and to send serial commands to the imager using text or macros.
- Click the **Utilities** button to show the tabbed interfaces for Differences from Default and Firmware.

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



Step 7 — Save Changes 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.
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 MS-2D.

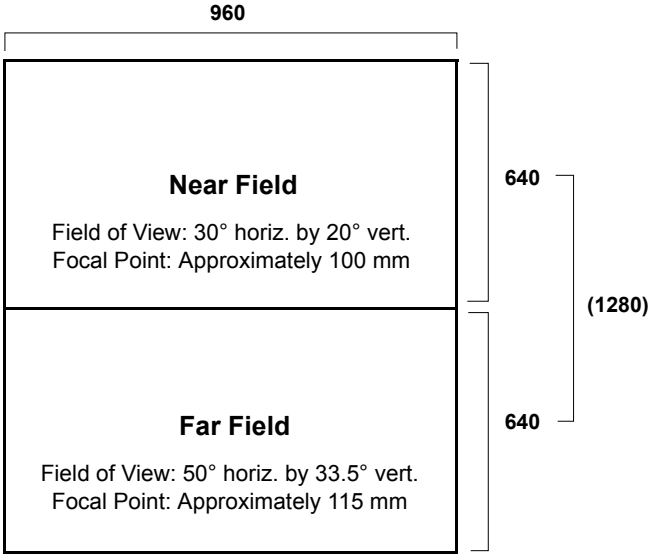
Saving Options

- **Send, No Save**. Changes will be lost when power is re-applied to the MS-2D.
- **Send and Save**. This activates all changes in current memory *and* saves to the MS-2D for power-on.
- **Send and Save, Including Factory**. This activates all changes in current memory, saves to the MS-2D for power-on, and saves the **Communications Mode** setting.

Dual Optics

The MS-2D's dual field optical system can read small 2D symbols as well as larger 1D symbols. It decodes the near and far fields simultaneously. The near field lens is for smaller symbols and the far field lens is for larger symbols. Move the MS-2D closer to read smaller symbols and farther away to read larger symbols.

Imaging Area



The MS-2D's optics are divided into far and near field decode zones. Each decode zone is 960 x 640 pixels.

Command Defaults

The following table shows each MS-2D command and its default setting.

Command	Default Setting
Communications	
Communications Mode	USB Native (HID)
Reader Packet Format	Raw
Reader to Host Packet Size	16384
Expect Host Response	Disabled
Reader Send Retry Count	3
Host Acknowledgement Timeout	15
Text Commands	Disabled; enable magic sequence
USB Keyboard Rate	10
Keyboard Mapping	US English (without leading 0 in alt-num)
Baud Rate	115.2K
Parity	None
Stop Bits	One
Data Bits	Eight
Text Command Timeout	11000
Read Cycle	
External Trigger	Read Both Fields
Default Continuous Event	Idle
Event Delay	100
Maximum Decodes per Read	1
Read Cycle Timeout	500
Ignore Duplicate Symbol Timeout	0
Targeting Zone Tolerance	1600
Morphological Preprocessing	None
Morphological Preprocessing Size	Small
AGC Sampling Mode	Automatic
Illumination	0
Exposure	25
Gain	11
AGC Frame Adjust Count	0
Symbologies	
Data Matrix	Enabled
QR Code	QR and Micro QR Code
Aztec Code	Disabled
Code 39	Enabled
Code 39 Checksum	Disabled
Code 39 Extended Full ASCII	Disabled

Command	Default Setting
Code 128	Enabled
BC412	Enabled
Code 93	Enabled
Codabar	Enabled
Codabar Checksum	Disabled
Interleaved 2 of 5	Enabled
Interleaved 2 of 5 Checksum	Disabled
Interleaved 2 of 5 Length	6 Digit Minimum
UPC	Enabled
EAN Status	Enabled
UPC-E as UPC-A	Disabled
Postal	Disabled
Pharmacode	Disabled
Pharmacode Fixed Symbol Length	Disabled
Pharmacode Symbol Length	5
Pharmacode Minimum Bars	4
Pharmacode Bar Width Status	Mixed
Pharmacode Direction	Forward
Pharmacode Fixed Threshold Value	10
GS1 DataBar	Enabled (All)
PDF417	Enabled
MicroPDF417	Disabled
Composite	Disabled
Composite Maximum Decodes per Read	1
Symbology Identifier	Disabled
I/O Parameters	
No Read Notification	Disabled
Targeting	Enabled
Beeper Volume	100
Beeper Duration	100
Beeper Separation	100
Beep on Good Read	Enabled
Button Stay-Down Time	0
Motion Detect Event	Read Both Fields
Motion Sensitivity	5
Motion Detect Start Delay	0
Image Quality	50

2 Using ESP

Contents

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 three configuration menus (**Communications**, **Read Cycle**, and **Symbologies**), a **Terminal** interface, and a **Utilities** interface.

ESP can be used to configure the MS-2D Engine in three main ways:

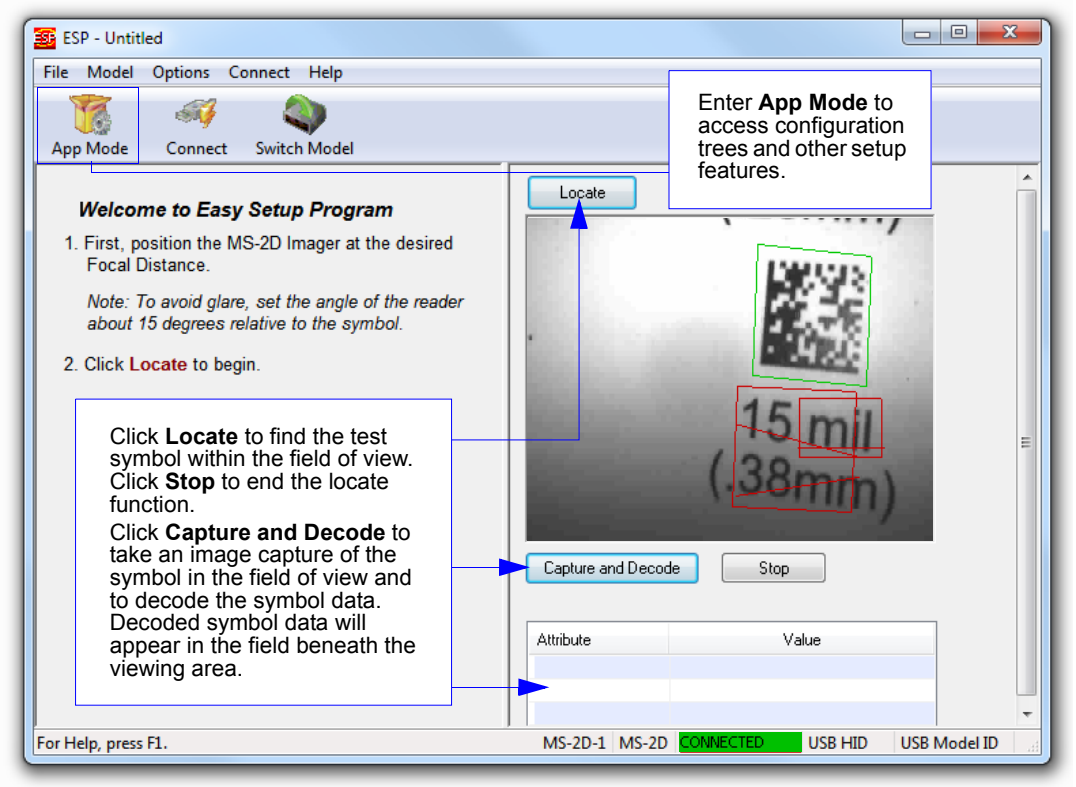
- **Tree Controls:** Each configuration menu contains a list of all option settings that pertain to that specific element of MS-2D operation. For example, the **Communications** menu shows a **Host Port Connections** option, and then a list of the sub-options **Baud Rate**, **Parity**, **Stop Bits**, and **Data Bits**. Each of these sub-options is configurable by using dropdown menus.
- **Graphic User Interfaces:** MS-2D 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 MS-2D 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**.

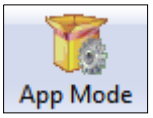
EZ Mode

EZ Mode offers instructions on positioning the MS-2D in relation to a test symbol, and also features a **Locate** and **Capture and Decode** function.

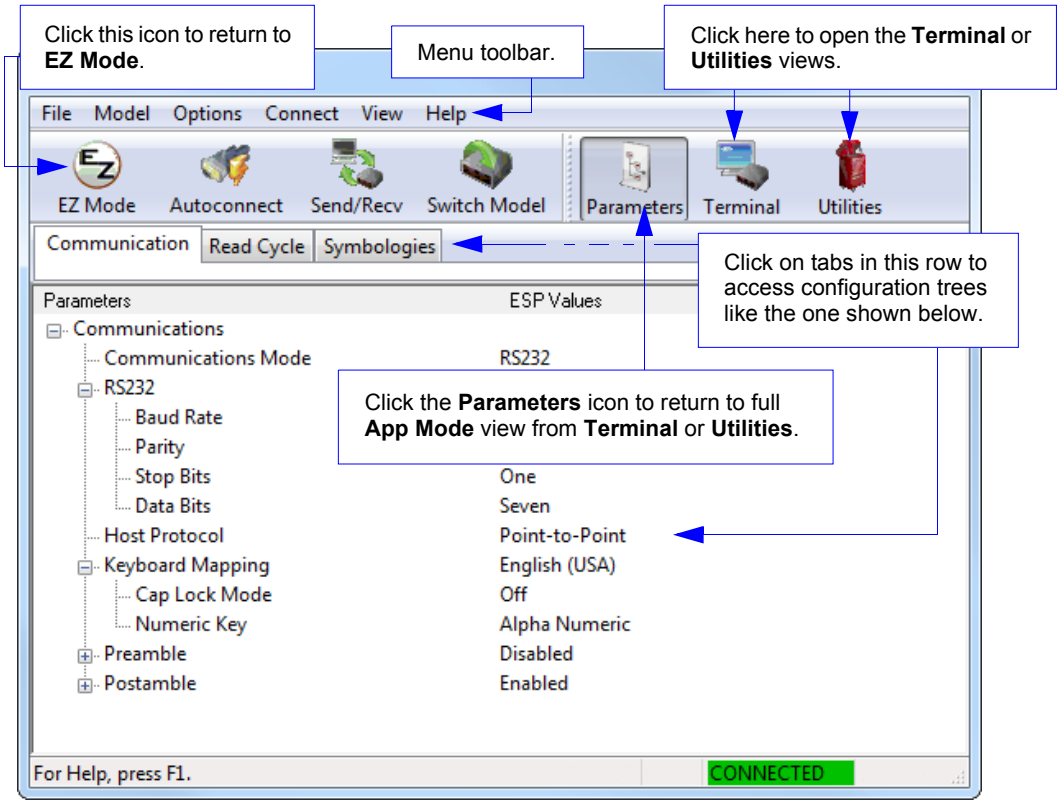


Application Mode

From **EZ Mode**, you can click on the **App Mode** button to access specific configuration menus, **Utilities** tools, 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.

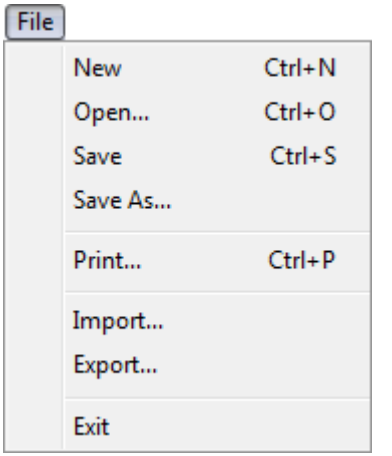


Note: See the corresponding sections of this documentation for specific information on any of the views or modes mentioned above.

Menu Toolbar

File > New

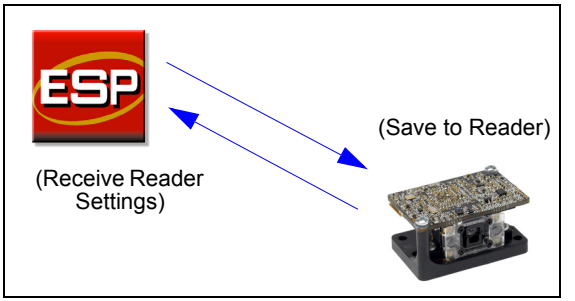
Whenever **New** is selected, the default configuration of **ESP** is loaded.



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 the MS-2D. The illustration below shows how settings can be saved and received between **ESP** and the MS-2D, and **ESP** and the host hard drive.



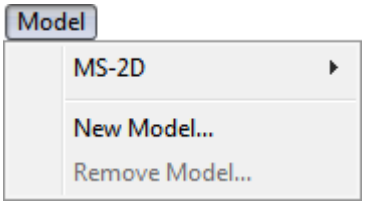
Import/Export

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.



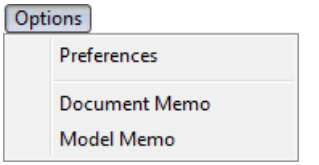
To connect to another model, select **New Model**, choose a new model from the pop-up menu that appears, and click **OK**.

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

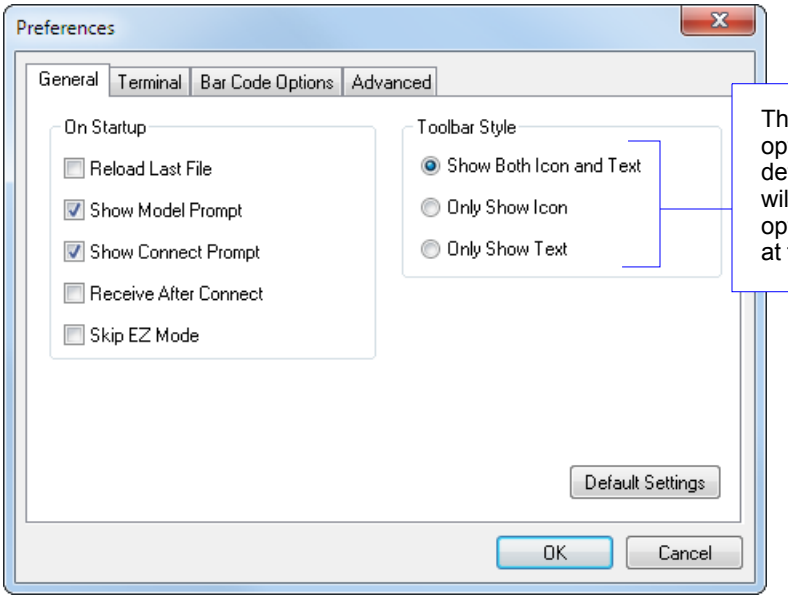
Options

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 MS-2D?** prompt.

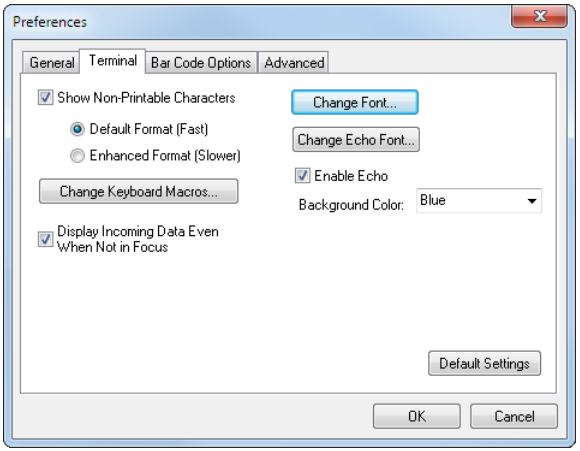
Receive After Connect

At startup, loads the MS-2D'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**.

Preferences > Terminal Tab

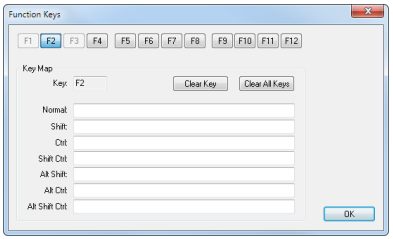


Show Non-Printable Characters

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 MS-2D on the Terminal screen.

Change Echo Font

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

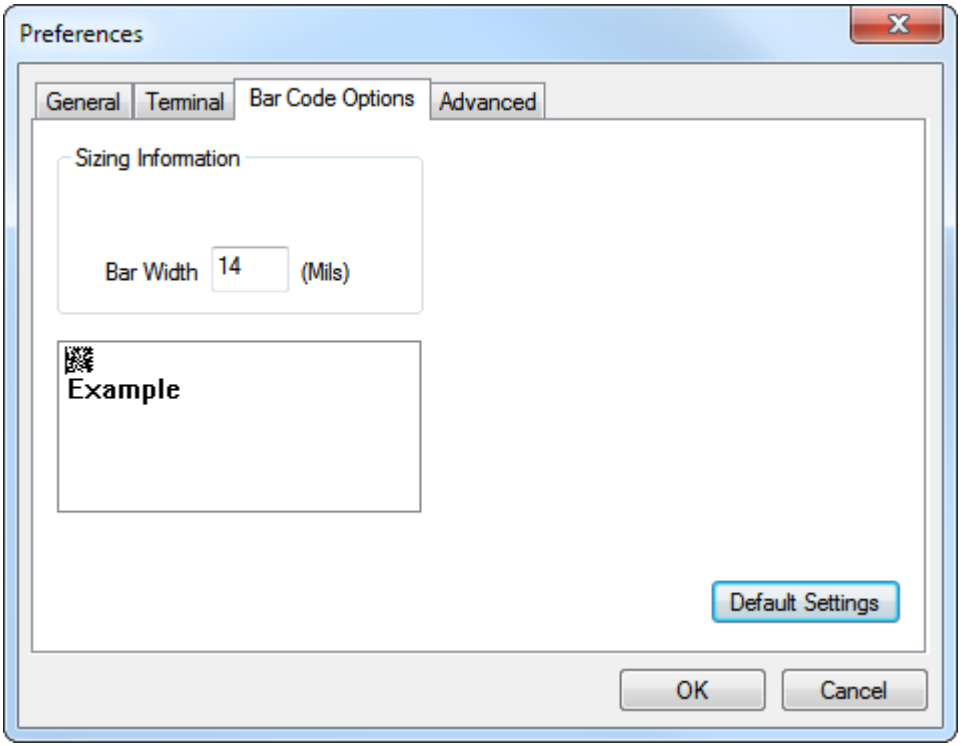
Enable Echo

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 MS-2D will continue to appear in the Terminal even when **ESP** is not the top window.

Preferences > Bar Code Options Tab



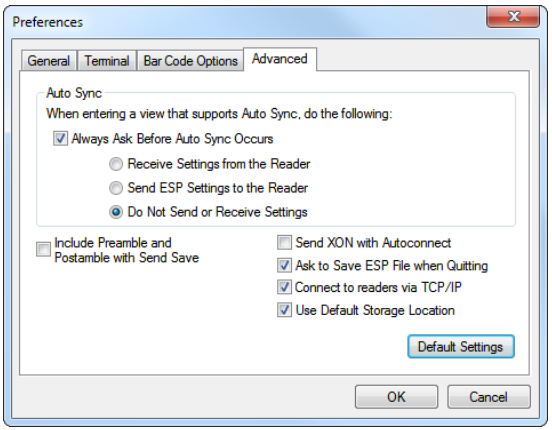
The **Bar Code Options** dialog allows you to set the size of user-created symbols.

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 14 is 0.014 inches.

Preferences > Advanced Tab



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

Always Ask Before Auto Sync Occurs

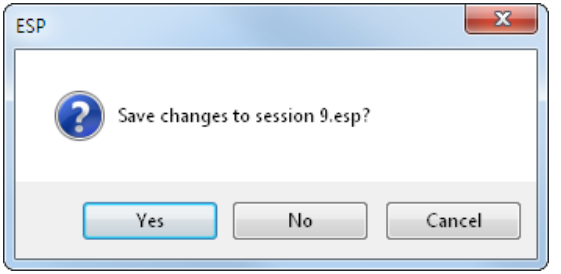
If this option box is checked, specific Auto Sync functions can be enabled. **Receive Settings from the Reader** will automatically send the imager's settings to **ESP** when Auto Sync is enabled. **Send ESP Settings to the Reader** will automatically send all imager configuration settings chosen in **ESP** to the imager. **Do Not Send or Receive Settings** creates a condition in which Auto Sync will not automatically send imager settings to **ESP**, or send **ESP** settings to the imager.

Send XON with Autoconnect

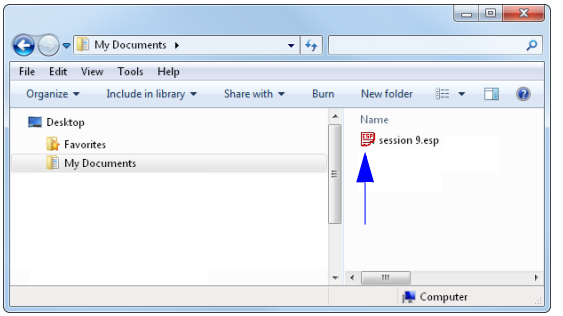
Sends an **XON** (**Begin Transmission**) command to the MS-2D before starting the **Autoconnect** routine.

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.

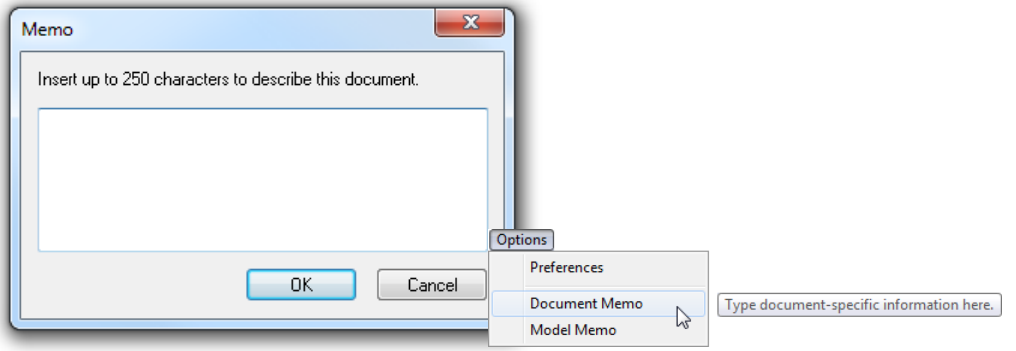


Use Default Storage Location

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

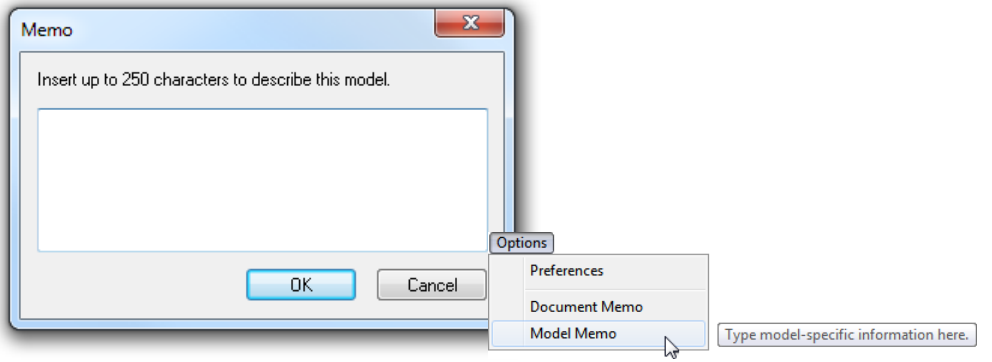
Document Memo

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

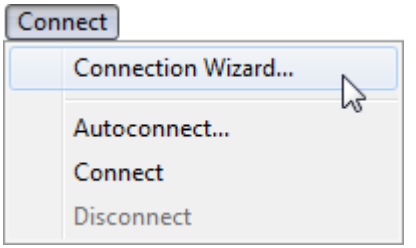
Similar to **Document Memo**, the information you type in the **Model Memo** field will appear in a context-sensitive text box whenever your cursor hovers over the **Model Memo** item on the **Options** menu. Memos created in **Model Memo** are specific to the model enabled when the message was created.



Note: Memos must be saved in a **.esp** file if you want them to 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

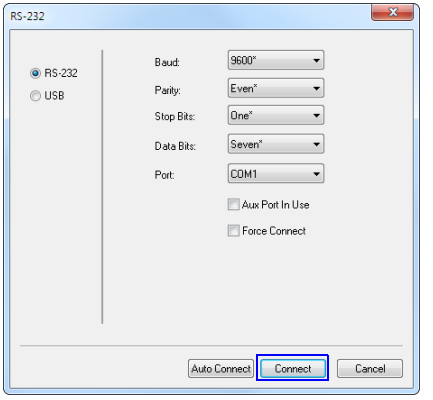
The **Connect** dropdown menu allows the user to access the **Connection Wizard**, as well as the **Autoconnect** and **Configure Multidrop** dialogs. **Connect** and **Disconnect** can also be performed directly from the dropdown menu without opening a dialog.



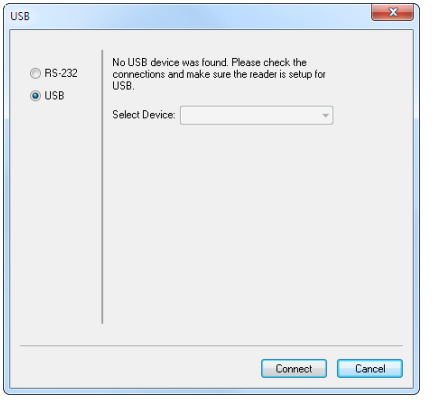
Connection Wizard

To connect using the Connection Wizard:

- Click **Connect** on **ESP**'s menu toolbar, and then select **Connection Wizard**.
- Select **RS-232** or **USB** to activate the appropriate display.
- Configure RS-232 or USB settings as required by the application, and click **Connect**.



RS-232 Connection Wizard



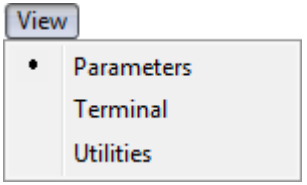
USB Connection Wizard

- 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 MS-2D and the host.

View

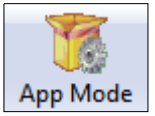
The **View** menu allows you to move quickly between interfaces without using the icon buttons on the **App Mode** toolbar.



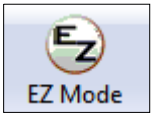
Notice that each menu item corresponds with the icon buttons at the top of the ESP window.

Navigating in ESP

To change MS-2D settings, or to access the **Terminal** or **Utilities** views, click the **App Mode** button.

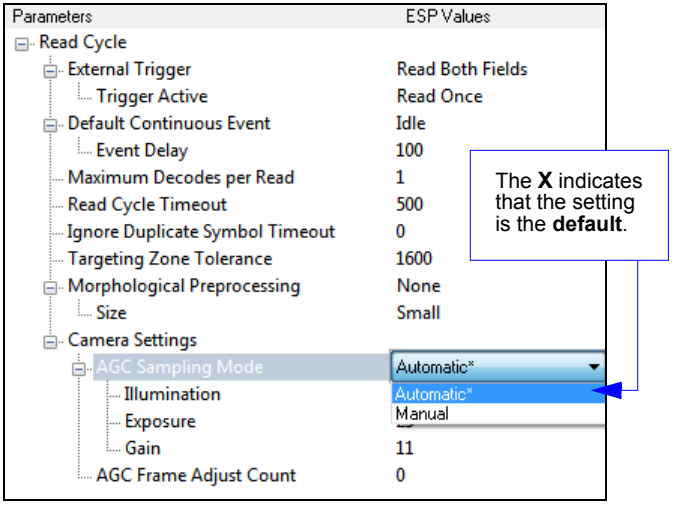


To return to EZ Mode, click the **EZ Mode** button.

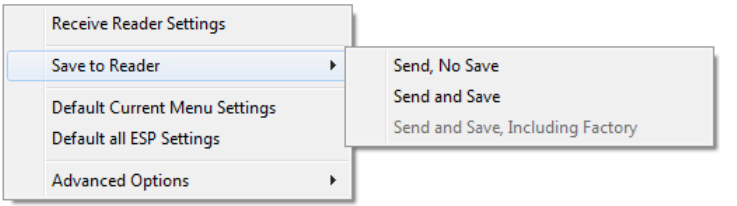


To make changes to configuration settings in the tree controls:

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 MS-2D. You can send the command without saving it, you can send and save the command simultaneously, or you can send and save the command and modified factory communications defaults 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

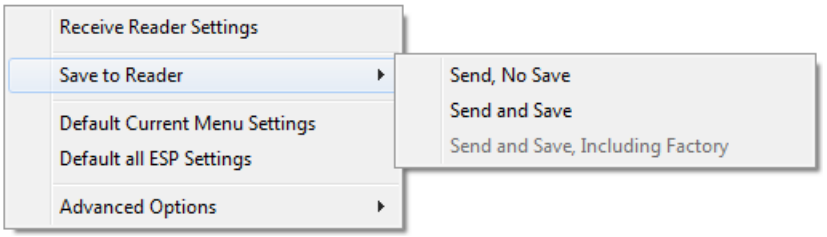
From the **Send/Recv** menu, select **Receive Reader Settings**.

Caution: Do not select this option if you do not want to upload the MS-2D's settings. For example, if your **ESP** file has a number of custom settings that you want to maintain and download into the MS-2D, these settings would be lost by choosing **Yes**.

This is useful if you want to receive (upload) the MS-2D's settings and save them as a file for future use. For example, if your MS-2D 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 MS-2D'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 MS-2D for power-on.

Send and Save, Including Factory (<Zp>)

Activates all changes in current memory, saves to the MS-2D for power-on, and saves the **Communications Mode** setting.

Important: Enable Send and Save as Factory Settings must be selected in General Preferences for this command to function.

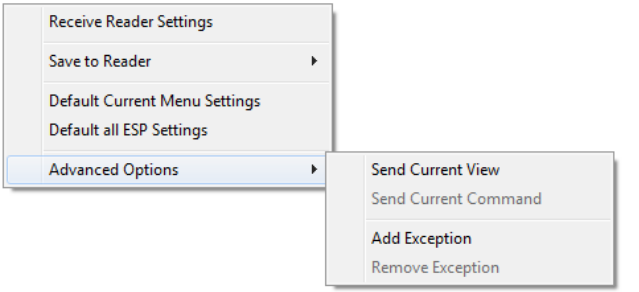
Defaulting

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 tree control are sent.



Send Current Command

Saves only the command that is currently selected in the tree control.

Add/Remove Exception

After you perform a **Receive Reader Settings** command[1] and you click on the **Advanced Options > Add Exception** option, you may see a list of serial commands. These are commands that may be in your MS-2D's firmware but are not included in (or are different from) your current version of **ESP**. When exceptions are present, the **Exceptions** button will appear to the right of the other mode buttons (**Communication**, **Read Cycle**, **Symbologies**). When no exceptions are present, the button will disappear.

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

It is important to note that these commands will be saved to your MS-2D whenever you send a **Save to Reader** command, or an **<A>** or **<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.

3 Integration

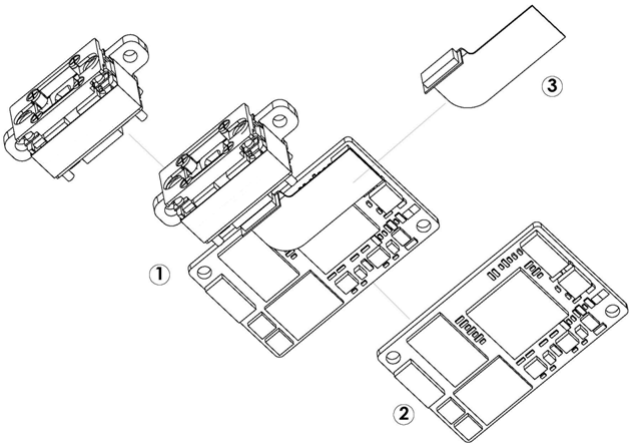
Contents

This section describes how to integrate the MS-2D mechanically and electrically.

Mechanical Specifications

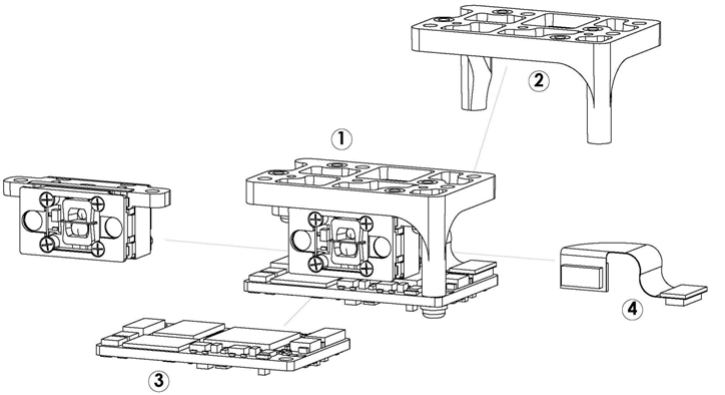
2D Engine with Decode Board Components

1. Fully-assembled MS-2D Engine
2. Decode Board
3. Decode-to-Engine Flex Cable



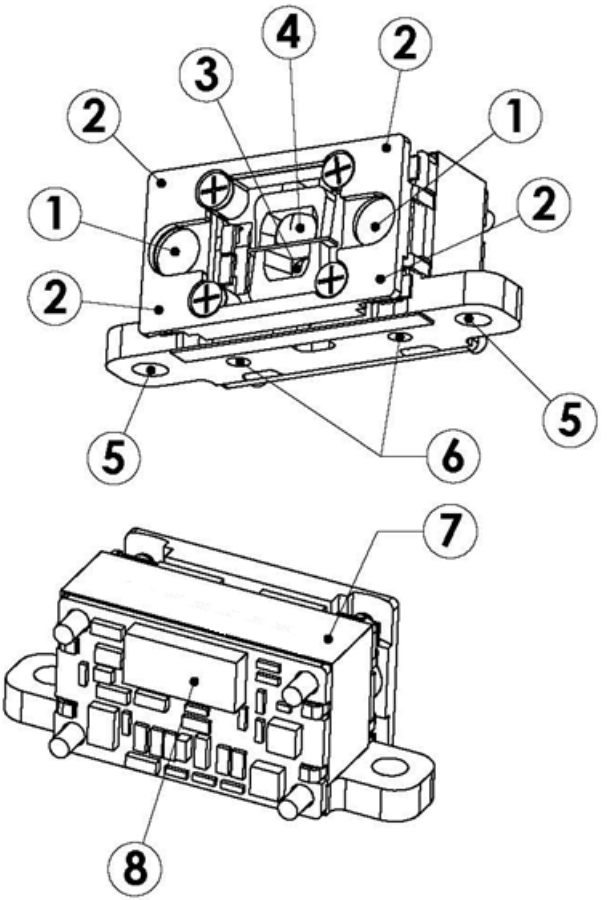
2D Engine with Mounting Bracket Components

1. Fully-assembled MS-2D Engine with Mounting Bracket
2. Mounting Bracket (has multiple possible configurations)
3. Decode Board
4. Decode-to-Engine Flex Cable



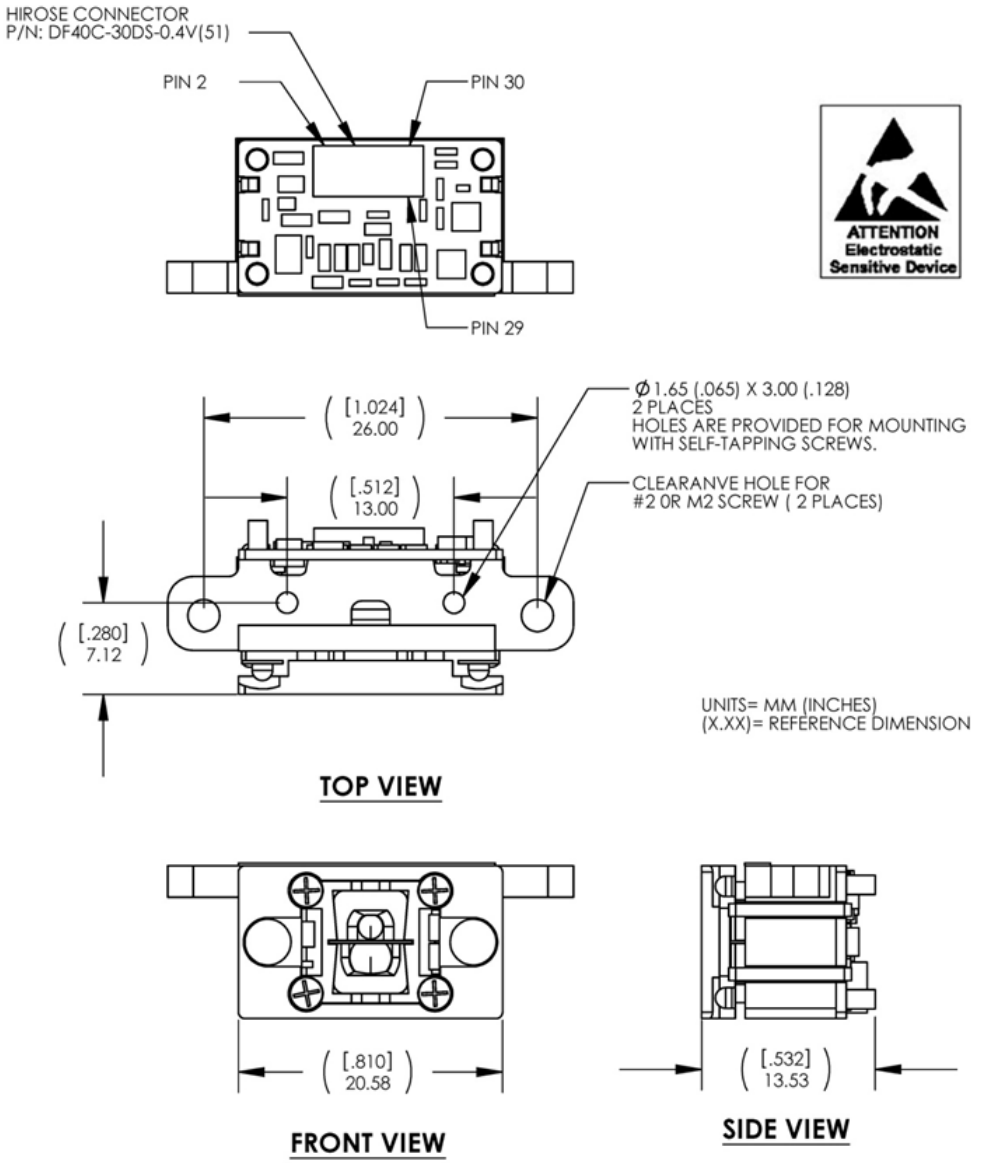
2D Engine with Mounting Tab Components

1. Blue LED Targeting Lens
2. Red LED Illumination Lens
3. High Density Field Lens
4. Wide Field Lens
5. Mounting Tabs
6. Self-tapping Screw Tabs
7. Printed Circuit Boards
8. Flex Cable Connector (connects Engine to Decode Board), Receptacle, 30 pin, 0.4 mm Pitch



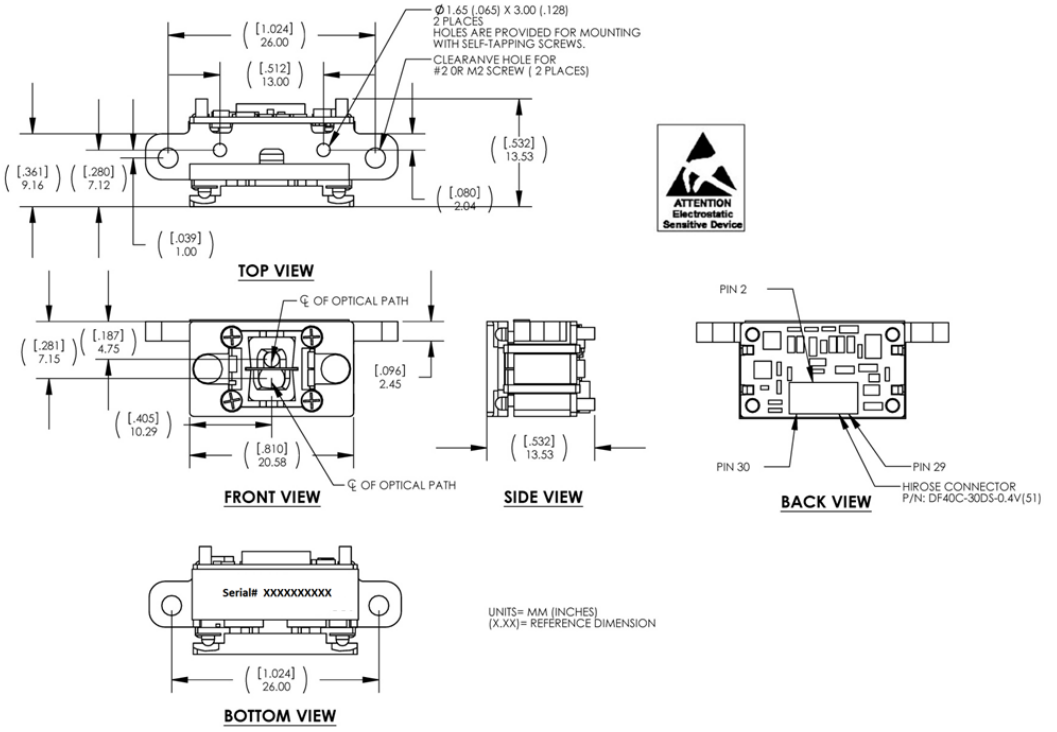
2D Engine Mechanical Specifications

1. The MS-2D has two holes available for mounting with two self-tapping screws.
2. Use M2.2 x 4.5 Philips pan head, type AB, steel, zinc clear, trivalent self-tapping screws.



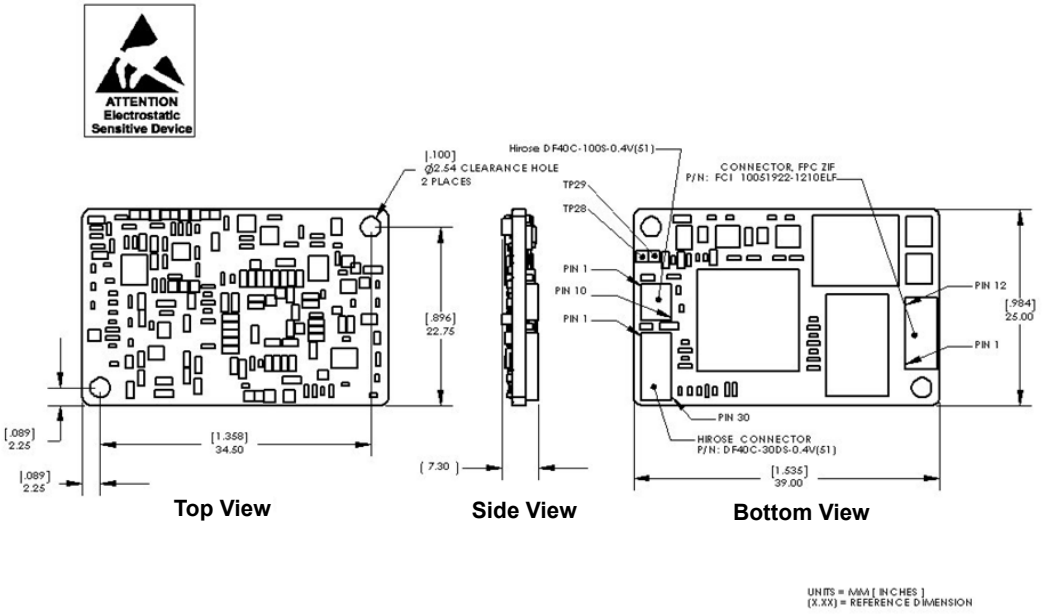
2D Engine with Mounting Tabs Mechanical Specifications

1. The MS-2D with mounting tabs has two tabs with mounting holes as well as two holes available for mounting with two self-tapping screws.
2. For the mounting tabs, use M2.2 x 6 Philips pan head screws. The design does not require a washer, but if a washer is desired, Omron Microscan recommends a flat washer, No. 2 screw size, .19” OD, .01” - .03” thick.
3. For the blind holes, use M2.2 x 4.5 Philips pan head, type AB, steel, zinc clear, trivalent self-tapping screws.



Decode PCB Mechanical Specifications

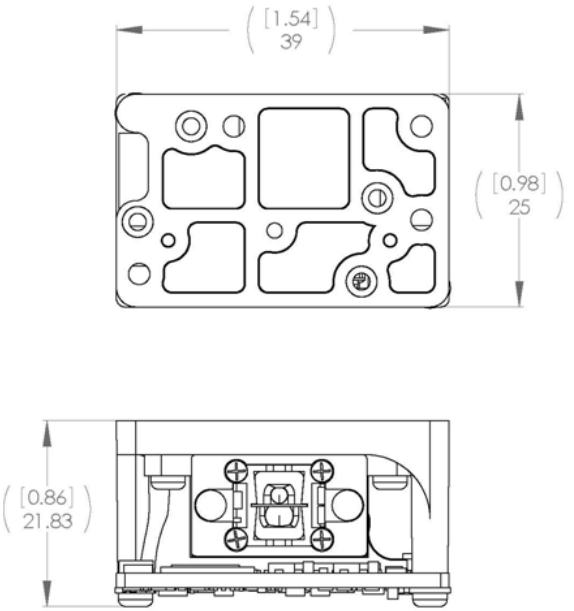
1. The MS-2D Decode PCB has two mounting holes.
2. Use M2.2 mounting hardware.



MS-2D Engine with Bracket Specifications

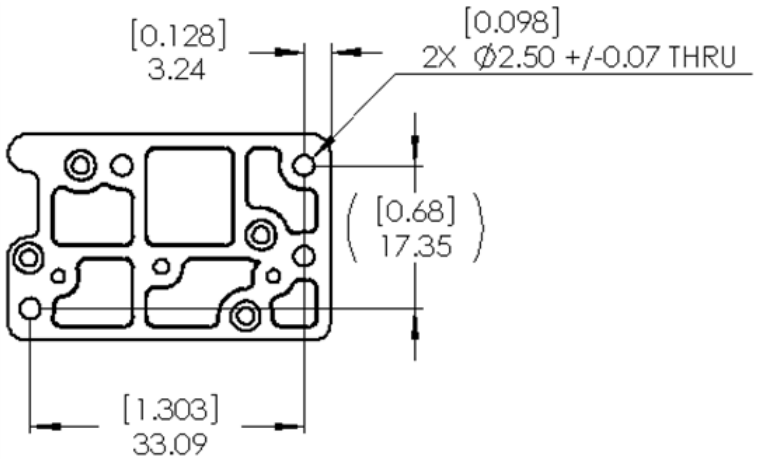
The MS-2D bracket has six holes for mounting the device: two unthreaded holes and four threaded holes. This allows the use of both self-tapping and machine screws in the target application.

Overall Dimensions



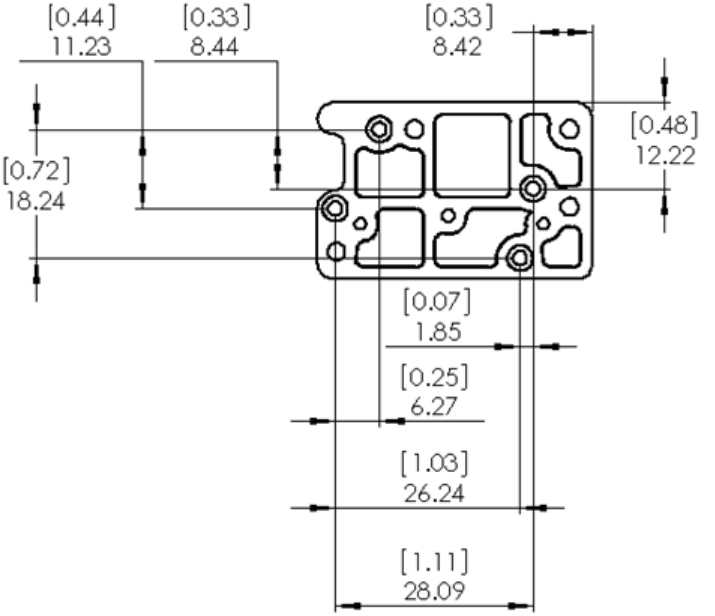
Non-Threaded Mounting Holes

There are two 2.46 mm diameter non-threaded mounting holes on the MS-2D bracket, shown here:



Threaded Mounting Holes

There are four M2 threaded mounting holes on the MS-2D bracket, shown below. Omron Microscan recommends using a M2x5 machine screw for mounting.



Enclosure Specifications

The MS-2D is a Class B device per EN 55022:2006 and complies with EMC Directive 2004/108/EC.

- Minimize infiltration by airborne contaminants and foreign materials while allowing sufficient air flow to maintain safe temperatures.
- The MS-2D must not come in contact with water.
- The MS-2D is sensitive to Electrostatic Discharge (ESD) and must be handled appropriately. Any individual that handles the MS-2D should be grounded using a wrist strap and ESD protected work area and work surface.
- The warranty of the MS-2D is void if the recommendations above are not followed when handling or integrating the device.

Optical Considerations

Window Requirements

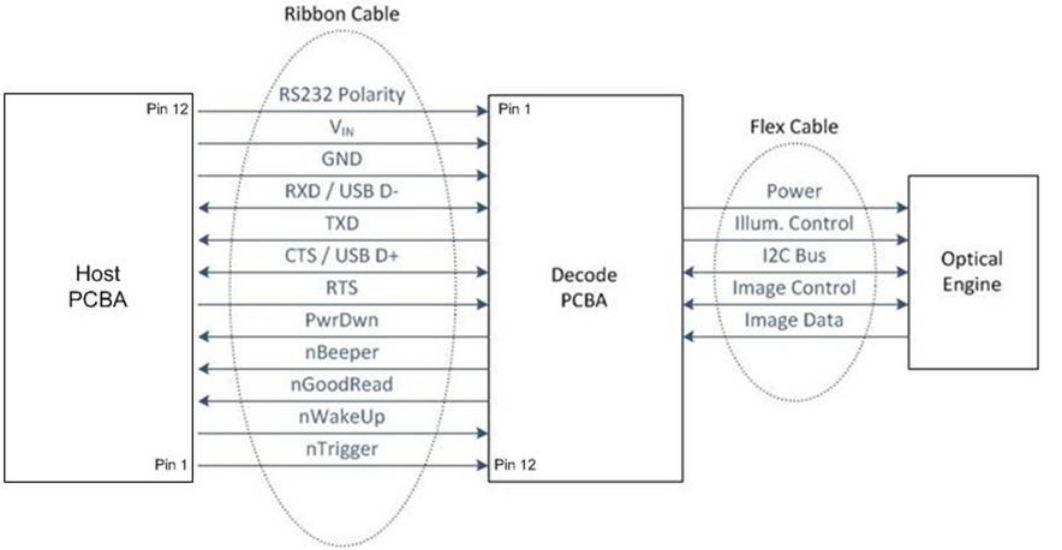
When integrating the MS-2D into a device or application, it may be necessary to install a window in front of the optics of the 2D Engine. Although many different types of material can be used, Omron Microscan recommends the following:

- Placement: Contact to 0.5 mm from the face of the 2D Engine.
- Material: Optically clear acrylic.
- Thickness: 1 mm.

- Contact Omron Microscan for additional information.

Electrical Specifications

The MS-2D Engine is a complete system that is easy to integrate into any device. The block diagram below shows the main components of the system. The MS-2D includes an optical engine with dual field optics (Wide Angle Field and High Density Field are both included in the standard product.) The MS-2D also includes a decode board that includes a microprocessor with embedded decode algorithms and a communication interface. The 12-inch ribbon cable connects the decode board to the host application. A 2-inch or 6-inch ribbon cable can be purchased separately.



System Requirements

Power Supply: The MS-2D is powered from the host via the VIN and GND pins. VIN must be within the range specified when measured at the decoding board. VIN must be maintained with varying loads, such as when the illumination is turned ON and OFF.

Host Ribbon Cable: The impedance of the cable for the USB data lines should be 90 ohm differential. For 3.3V operation (RS-232 models only), a flex cable of no more than 6" in length can be used with a 0.28mm (0.011") trace width and 0.3mm trace thickness.

Power Sequencing: There is no special power sequence needed for the MS-2D as long as the max and min voltage and current specifications are met. However, if the voltage on a pin is greater than VIN, such as when powering on, then current will flow from the pin to VIN through the pull-up resistors.

Thermal Requirements: The operating temperature range for the MS-2D is 0º C – 70º C (32º F – 158º F) as measured at the coldest/hottest point on either of the printed circuit boards.

Boot Modes

The MS-2D will enter the following boot modes based on the configurations of the External Wakeup line and TP28: Illumination I/O 4 line during boot time. TP28 has an internal pull-up and can be pulled down by shorting it to TP29, which is a ground point located next to TP28.

Pin	Name	Type	Description
TP28	Illumination I/O 4	Bidirectional	Illumination Communications Line, Boot Select Signal
TP29	Ground	Power	Power supply and signal ground

Note: Pin has a weak pull-up to internal 3.3V.

Boot Mode	Illumination I/O 4	External Wakeup	Description	Note
1	Pulled-up	Pulled-up	Normal Mode – Reader Application	
2	Pulled-up	Pulled-down	Upgrade Mode - Linux Application to upgrade Reader Application	
3	Pulled-down	Pulled-up	Factory Restoration Mode	1
4	Pulled-down	Pulled-down	Uboot Upgrade Mode - Upgrading Linux OS and FS	

Note 1: Boot Mode 3 will clear all registers and any customer-specific configuration. Use with caution.

Host Interface Pinouts (RS-232)

Pin	Name	Type	Description	Note
1	RS-232 Polarity	Input	RS-232 polarity control. When high, all RS-232 signals have their normal polarity. When low, all RS-232 signals have inverted polarity.	1
2	VIN	Power	Power supply voltage input	
3	GND	Power	Power supply and signal ground	
4	RxD	Input	RS-232 receive data, TTL level	1
5	TxD	Output	RS-232 transmit data, TTL level	1
6	CTS	Input	RS-232 Clear to Send, TTL level	1
7	RTS	Output	RS-232 Request to Send, TTL level	1
8	PwrDwn	Output	Power down indicator	1
9	nBeeper	Output	Active low signal; can be used to indicate errors or success.	1
10	nGoodRead	Output	Indicates a successful decode; active low	1
11	nWakeUp	Input	Bring the unit out of sleep state; active low	1
12	nTrigger	Input	Activate image acquisition, decode; active low	1

Note 1: Pin has a weak pull-up to VIN.

Host Interface Pinouts (USB)

Pin	Name	Type	Description	Note
1	<unused>			1
2	VIN	Power	Power supply voltage input	
3	GND	Power	Power supply and signal ground	
4	D-	Bidirectional	USB D- signal	
5	<unused>			1
6	D+	Bidirectional	USB D+ signal	
7	<unused>			1
8	PwrDwn	Output	Power down indicator	1
9	nBeeper	Output	Active low signal; can be used to indicate errors or success.	1
10	nGoodRead	Output	Indicates a successful decode; active low	1
11	nWakeUp	Input	Bring the unit out of sleep state; active low	1
12	nTrigger	Input	Activate image acquisition, decode; active low	1

Note 1: Pin has a weak pull-up to VIN.

Electrical Control Signals

The MS-2D is equipped with various control signals (lines or pins) that control or provide output on a variety of functions with the device. See the MS-2D Timing Diagrams and Timing Table in this guide for additional details.

Pin 8 - Power Down. The PwrDwn line will be asserted HIGH when VIN is present and the MS-2D has switched to the sleep state. PwrDwn will be transition to the LOW state when VIN is not present or when VIN is present and the MS-2D is not in sleep state.

Pin 9 - Beeper. The nBeeper line can be pulled LOW for a specified length of time or for a series of pulses of a specified duration on a successful decode or on certain error conditions. The length or type of signal can be determined via a configurable register.

Pin 10 - Good Read. Upon a successful scan and decode, the nGoodRead line will be asserted LOW. The length of nGoodRead assertion can be set via a configurable register.

Pin 11 - Wakeup. Once the MS-2D has entered the sleep state, it may be awakened by asserting nWakeUp LOW. Note that nWakeUp must be HIGH when the MS-2D enters the sleep state in order for nWakeUp to awaken the MS-2D on assertion. The MS-2D awakens to the idle state. Additionally, as long as the nWakeUp line is held low, it will prevent the MS-2D from entering the sleep state.

Pin 12 - Trigger. To activate the MS-2D, pull the nTrigger line LOW.

Note: All output signals except USB D- and USB D+ are connected to open drain buffers with a pull-up of 100 Kilo-ohm to VIN and a maximum current capability of 50 mA. All input signals except USB D- and USB D+ are connected to a pull-up to VIN and to a buffer with a 50 mA maximum current capability.

Note: When VIN is initially supplied, PwrDwn and the other outputs will be LOW for a few milliseconds until the voltages on the board come up. They will then transition to default HIGH due to pull-ups until the unit is up and running. These signals should be ignored until the unit is fully functioning.

Power Modes

Boot State. The MS-2D enters boot state upon application of VIN. The PwrDwn pin will be HIGH (after power on delay) until the main app starts.

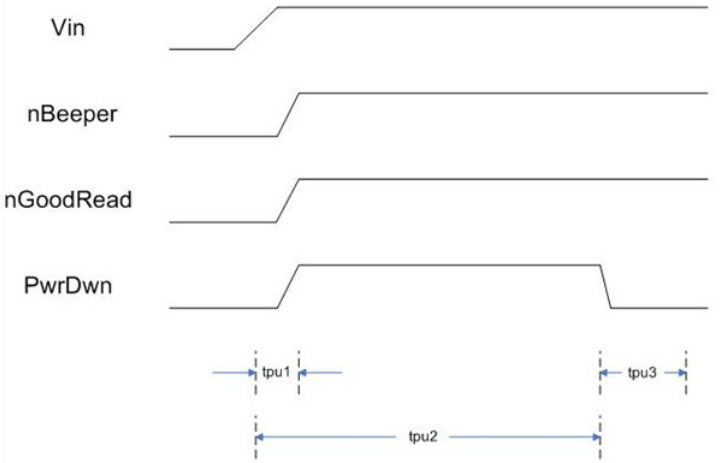
Active State. In the Active state the unit is capturing images and initiating the decode process and/or storing images. The unit transitions to Active state from Idle state when a trigger event is received.

Idle State. In the Idle state the unit is not actively capturing images. The processor is fully functioning and communication can take place, upgrades can be performed, and scripts can be run. The Idle state is entered from the Boot state after power on, from the Active state after a register defined timeout in which there are no trigger events, and from the Sleep state on receipt of a wake up.

Sleep State. The imager, illumination, and most of the processor is powered down. The CPU wake up circuitry, the memory, and the input/output buffers are powered. The unit enters the sleep state after a register defined timeout of inactivity. On receipt of a wake up on the nWakeUp pin, the processor restores the run environment and enters the Idle State.

Power On / Boot Timing Diagram

The PwrDwn signal will transition to HIGH shortly after VIN is applied and will remain HIGH until the main application starts.

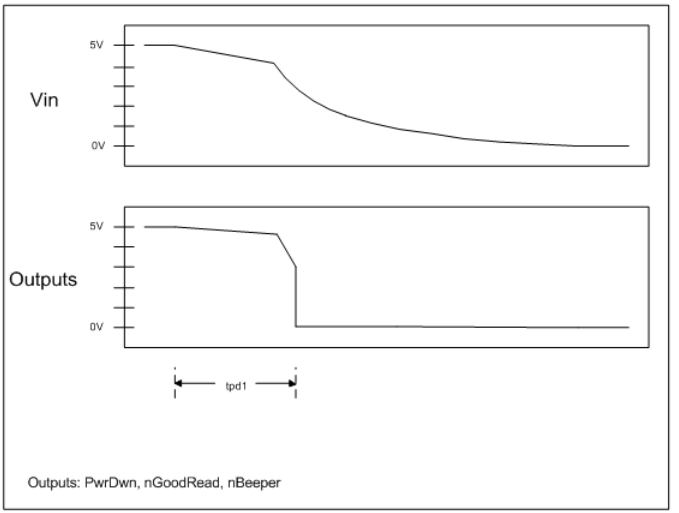


Parameter	Symbol	Min	Typ	Max	Unit	Note
Power on to outputs at default	tpu1		10		msec	
Power on to PwrDwn deasserted	tpu2		9		sec	
PwrDwn deasserted to ready (Idle state), after power on	tpu3		1		sec	1

Note 1: USB enumeration might take longer on USB model.

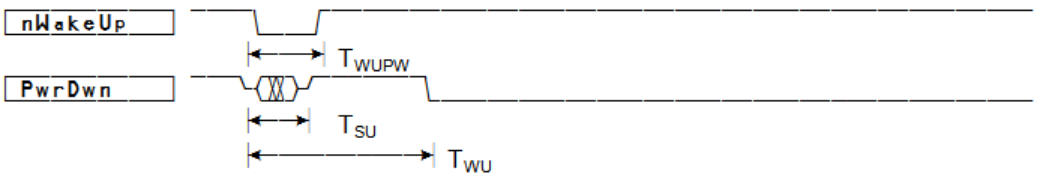
Power Down Timing Diagram

Power (VIN) can be removed at any time except when the unit is performing an upgrade. Removing power during an upgrade may cause the unit to become unusable.



Parameter	Symbol	Min	Typ	Max	Unit
Power off to outputs low	tpd1		56		msec

Sleep-to-Wakeup Timing Diagram



Signal	Description	Min	Typ	Max	Unit
TWUPW	nWakeUp pulse width	10	20		msec
TWU	Time between nWakeUp asserted and MS-2D ready			120	msec

Image Capture and Decode Timing Diagram

Image Capture and Decode. Image acquisition and decoding can be started from either the nTrigger line or via communications channel command. The time required to capture an image can vary depending on the size of image selected, the confirmation time register, and where the imager is in the capture cycle. The time to decode an image can depend on the image quality, complexity of the symbol, etc. The maximum time spent trying to decode an image is a register.

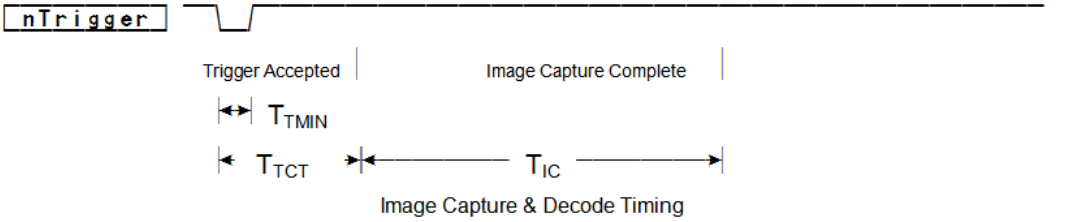


Image Capture & Decode Timing

Parameter	Symbol	Min	Typ	Max	Unit	Note
Trigger accepted to image capture complete	TIC	25		50	msec	1,2
Minimum trigger duration	TTMIN					1,4
Trigger confirmation time	TTCT	0		2^^31		5

Notes:

1. Confirmation time = 0.
2. Maximum image size.
3. Trigger must also be asserted for confirmation time.
4. Trigger must be asserted for minimum trigger duration even if trigger confirmation time is less than minimum trigger duration.
5. Trigger confirmation time is adjustable through user selectable settings.

2D Engine Ribbon Cable Diagram (Decode Board to Host Interface)

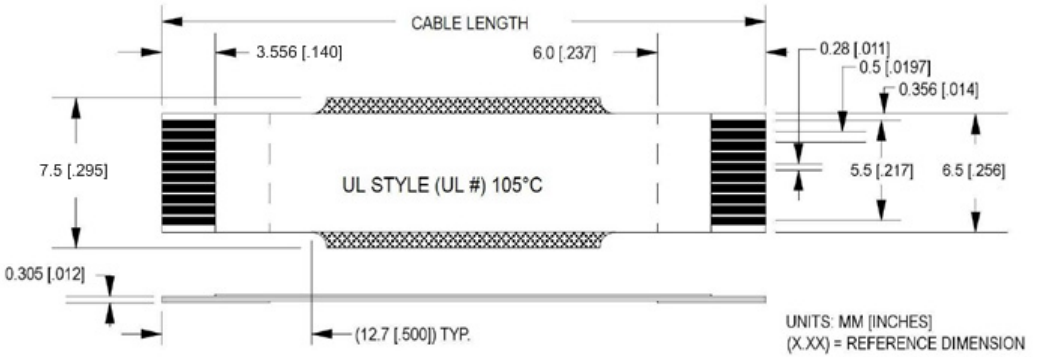
Our ribbon cables have the following characteristics:

- Bottom contact on MS-2D mating end
- 12 pin
- 0.5 mm pitch
- 0.3 mm thickness with stiffener.

Note that ribbon cables have contacts on the same side of each end. Take this into account with respect to the control signals when designing the mating connector pinout on the host interface.

3.3V operation of the MS-2D is only possible with 6” or shorter ribbon cable, and only with RS-232 models of the product.

Three ribbon cables are available with part numbers and lengths shown in the table below.



Part Number	Description
61-000209-01	Ribbon Cable, MS-2D, 2 in. (50.8 mm)
61-000209-02	Ribbon Cable, MS-2D, 6 in. (152.4 mm)
61-000209-03	Ribbon Cable, MS-2D, 12 in. (304.8 mm)

Electrical Characteristics (DC) – Absolute Maximum Ratings

Parameter	Symbol	Min	Max	Unit
DC Supply Voltage	VIN	-0.5	5.5	V
DC Input Voltage	VI	-0.5	5.5	V
DC Output Voltage	VO	-0.5	5.5	V
Output source or sink current	I/O		50	mA

Electrical Characteristics (DC) – Operating Conditions

Parameter	Symbol	Min	Typ	Max	Unit	Note
DC Supply Voltage	VIN	3.3		5.5	V	
High level input voltage	VIH	0.7 x VIN			V	4
Low level input voltage	VIL			0.8	V	4
High level output voltage	VOH			VIN		1
Low level output voltage	VOL			0.55	V	1
Output leakage current	IOZ			10	uA	
Inrush current	IP				mA	3
Active operating current	IA		345		mA	2, 5
Idle operating current	II		75		mA	
Sleep current	IS			1	mA	3
Maximum current available to expanded illumination circuitry	IXI				mA	
USB high level input voltage	VUSBIH	2.0			V	
USB low level input voltage	VUSBIL			0.8	V	
USB static output high	VUSBOH	2.8		3.6	V	
USB static output low	VUSBOL			0.3	V	
USB valid vbus voltage	VBUS	4.65		5.25	V	

Notes:

1. 100 Kilo-ohm pull-up to VIN on open drain output, actual voltage will depend on external impedance connected to pin.
2. Depends on the brightness level of the illumination LEDs.
3. Assumes inputs and outputs are tri-stated or high; if pulled low, current through pull-up resistors will need to be added.
4. Except USB D- and D+.
5. Continuous scan.

Decode PCB to 2D Engine PCB Connector

Pin	Name	Type	Description
1	VIN	Power	VIN power to Optical Engine
2	VIN	Power	VIN power to Optical Engine
3	1.8V	Power	1.8V power to Optical Engine
4	1.8V	Power	1.8V power to Optical Engine
5	GND	Power	Power and signal ground
6	ExtClk	Output	External clock to imager
7	GND	Power	Power and signal ground
8	PixClk	Input	Pixel clock
9	GND	Power	Power and signal ground
10	FrameValid	Input	Vsync from imager
11	LineValid	Input	Hsync from imager
12	GND	Power	Power and signal ground
13	Dout4	Input	Imager pixel data 4
14	Dout5	Input	Imager pixel data 5
15	Dout6	Input	Imager pixel data 6
16	Dout7	Input	Imager pixel data 7
17	Dout8	Input	Imager pixel data 8
18	Dout9	Input	Imager pixel data 9
19	Dout10	Input	Imager pixel data 10
20	Dout11	Input	Imager pixel data 11
21	nImagerReset	Output	Imager reset, active low
22	ImagerStandby	Output	Imager standby
23	Sdata	Input/Output	I2C bus data line
24	Sclk	Output	OutputI2C bus clock line
25	GND	Power	Power and signal ground
26	IllumPwm0	Output	PWM illumination signal
27	TargetLed	Output	Targeting LED control signal
28	1.8VImagerEnable	Output	Imager 1.8V enable
29	5VEnable	Output	Optical Engine 5V enable
30	2.8VEnable	Output	Optical Engine 2.8V enable

Decode PCB Expanded Illumination Connector

Pin	Name	Type	Description	Note
1	VIN	Power	Power supply voltage input	
2	VIN	Power	Power supply voltage input	
3	Illumination I/O 0	Bidirectional	Illumination Communications Line, UART Port4 TX TTL Level, I2C SDA	1
4	Illumination I/O 1	Bidirectional	Illumination Communications Line, UART Port4 RX TTL Level, I2C SCL	1
5	Illumination PWM 1	Output	Illumination PWM output	
6	Illumination I/O 2	Bidirectional	Illumination Communications Line, GPIO	
7	Illumination I/O 3	Bidirectional	Illumination Communications Line, GPIO	
8	Illumination I/O 4	Bidirectional	Illumination Communications Line, Boot Select Signal	1
9	GND	Power	Power supply and signal ground	
10	GND	Power	Power supply and signal ground	

Note 1: 10 Kilo-ohm pull-up to VIN 3.3V.

PCB Illumination Connector

Pin	Name	Type	Description
1	NC		
2	5V	Power	5V power to Illumination
3	IllumLed0	Output	Illumination LED0 control
4	GND	Power	Power and signal ground
5	NC		

Compliance Information

The MS-2D conforms to the following Product Specifications:

EMC Emissions:	EN 55022:2006 class B limits
LED Safety Compliance:	IEC 62471:2006 (Ed. 1.0)
ICES-003 Compliance:	ICES-003
FCC Compliance:	FCC Part 15, Subpart B

The product herewith complies with the requirements of the following Directive and carries the CE Marking accordingly:

- EMC Directive 2004/108/EC

This certification applies to the MS-2D as a stand-alone OEM engine and does not apply to the MS-2D as an integrated module. When integrating the MS-2D into another product, that product will still need to obtain any applicable CE certifications for the full product.

The MS-2D is RoHS compliant.

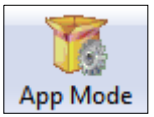
4 Communications

Contents

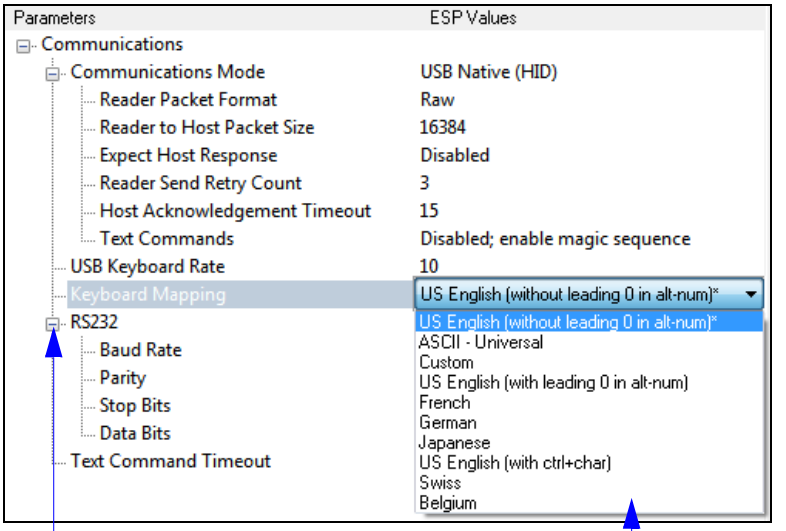
This section explains how to set up communications between the MS-2D and a host.

With Omron Microscan's **ESP** (Easy Setup Program), configuration changes can be made in the **ESP** tree controls and then sent and saved to the MS-2D. The Data Matrix symbols in this section can also be decoded to configure the MS-2D's Communications parameters.

Communications by ESP



Click this button to bring up the **App Mode** view, then click the **Communication** tab.

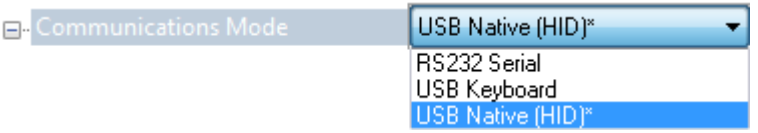


To open nested options, **single-click** the **+**.

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

Communications Overview

Whenever you default the MS-2D, it will return to the default settings of whichever interface you are using. Defaulting the imager does not remove preamble and postamble formatting. The MS-2D is in **USB Native (HID)** by default.



USB

With USB communications, the imager connects directly to the host's USB port from which it draws its power. Data is displayed by any open Windows-based program that can capture text in USB Keyboard Mode.

RS-232

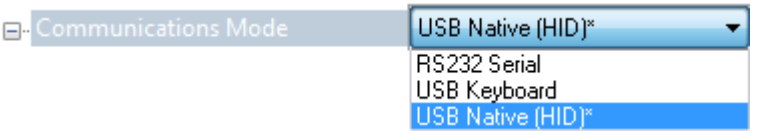
With RS-232 communications the imager communicates with the host through a communications program such as HyperTerminal.

Default settings for establishing RS-232 communications are:

Baud =	**115.2K**
Data Bits =	**8**
Parity =	**None**

USB Interface

The MS-2D is in **USB Native (HID)** by default.



USB Keyboard Mode (Windows)

Data is output as keyboard sequences.

M0002_01

USB Native (HID)

This mode is the standard way of transferring unformatted, unpacketized data to the imager through the USB port.

M0004_01

USB Virtual COM Mode

This mode allows an MS-2D in a USB configuration to function as a virtual serial COM port. This mode requires installation of a USB Virtual COM driver, which is available from Omron Microscan by request.

M0005_01

RS-232 Interface

Enabling either of these modes will disable USB communications and require you to default the imager or read the “USB Keyboard” symbol to return to USB.

Baud Rate (RS-232)

Baud Rate is the rate at which the imager and host transfer data. It only needs to be changed if necessary to match the host setting.


M0020_01

1200


M0024_01

19.2K


M0021_01

2400


M0025_01

38.4K


M0022_01

4800


M0026_01

57.6K


M0023_01

9600


M0027_01

115.2K (Default)

Parity (RS-232)

Parity is an error detection routine in which one data bit in each character is set to **1** or **0** so that the total number of 1 bits in the data field is even or odd. It only needs to be changed if necessary to match the host setting.


M0030_01

None (Default)


M0029_01

Odd


M0028_01

Even

Data Bits (RS-232)

Data Bits are the total number of bits in each character. This setting only needs to be changed if necessary to match the host setting.


M0019_01

8 Data Bits (Default)


M0018_01

7 Data Bits

Preamble

A **preamble** is a character or series of characters that is added to the beginning of a decoded data string. Preamble characters will appear in the order that they are enabled (left to right). For example, if you enable a comma and then a space, and then decode a symbol containing the data 'ABC', your output will look like this:

, ABC

The only limit to the number of preambles enabled is the total memory size available. Set the desired preamble by reading the appropriate symbol below.


M0130_02

Comma


M0131_02

Space


M0133_02

Tab (RS-232 Only)


M0132_02

Tab (USB Keyboard Only)


M0135_02

Carriage Return Line Feed (RS-232 Only)


M0134_02

Erase (None)


M0145_02

Erase Preamble and Postamble Data

Postamble

A **postamble** is a character or series of characters that is added to the end of a decoded data string. Postamble characters will appear in the order that they are enabled (left to right). For example, if you enable a space and then a comma, and then decode a symbol containing the data 'ABC', your output will look like this:

ABC ,

The only limit to the number of postambles enabled is the total memory size available.

Set the desired postamble by reading the appropriate symbol below.


M0137_02

Comma


M0140_02

Space


M0143_02

Tab (RS-232 Only)


M0142_02

Tab (USB Keyboard Only)


M0136_02

Carriage Return (RS-232 Only)


M0138_02

Line Feed (RS-232 Only)


M0139_02

Carriage Return Line Feed (RS-232 Only)


M0141_02

Enter (USB Keboard Only)


M0144_02

Erase (None)

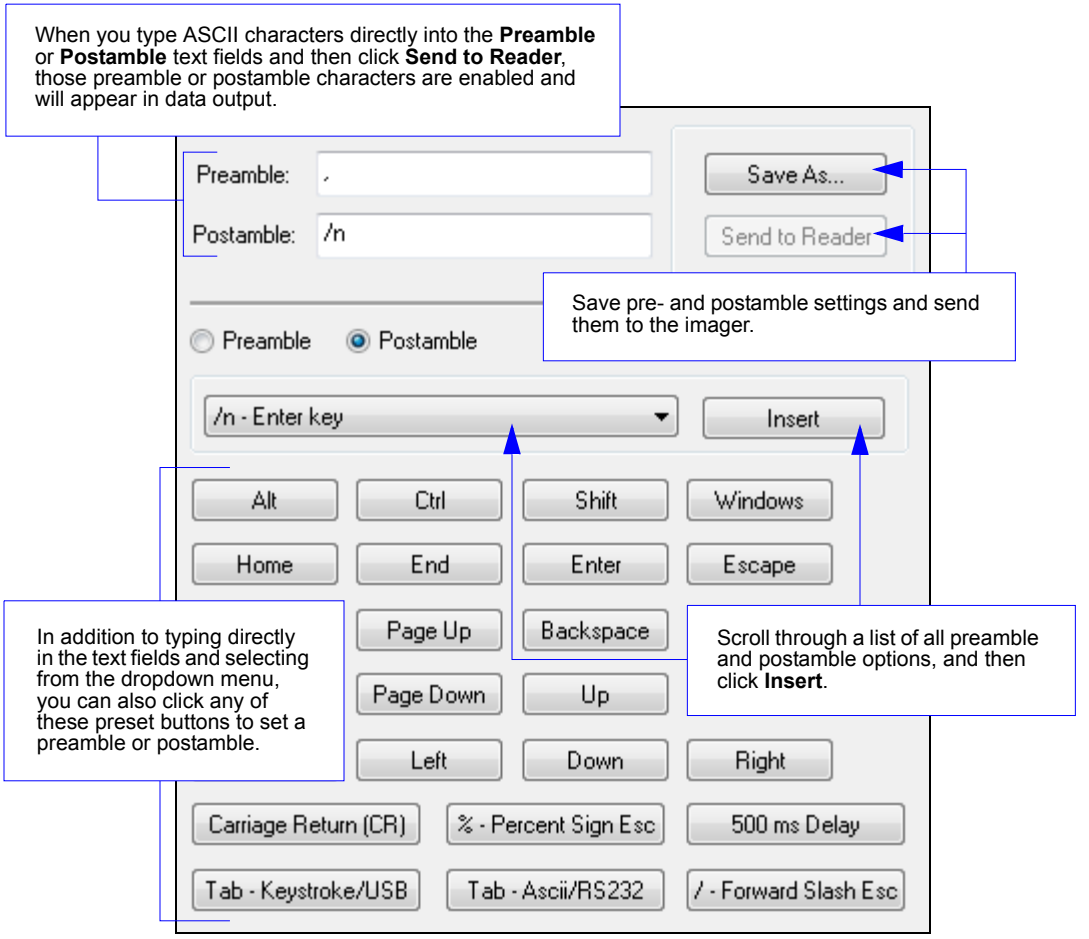

M0145_02

Erase Preamble and Postamble Data

Preamble and Postamble by ESP

Characters can also be added to the beginning and end of data strings using **ESP**. There are a few different ways to do this, using the interface shown below.

You will see the Communications tree control on the left, and the Preamble/Postamble interface on the right.



Keyboard Mapping

The **Keyboard Mapping** feature provides alternatives for keyboards that do not conform to U.S. English mapping.

Note: Universal Keyboard mapping is slightly slower than the other language-specific options, because it maps data by reference to the full set of ASCII characters. The advantage of Universal Keyboard mapping is that it allows any language and keyboard layout to be mapped.

Important: Keyboard Mapping is not to be confused with USB Keyboard Mode, which has an entirely different function—namely to enable USB cabled communications. (See **USB Interface**).


M0008_02
U.S., No Leading 0 (Default)


M0007_02
U.S. with Leading 0


M0009_02
U.S., Ctrl + Char.


M0010_01
French


M0011_01
German


M0012_01
Japanese

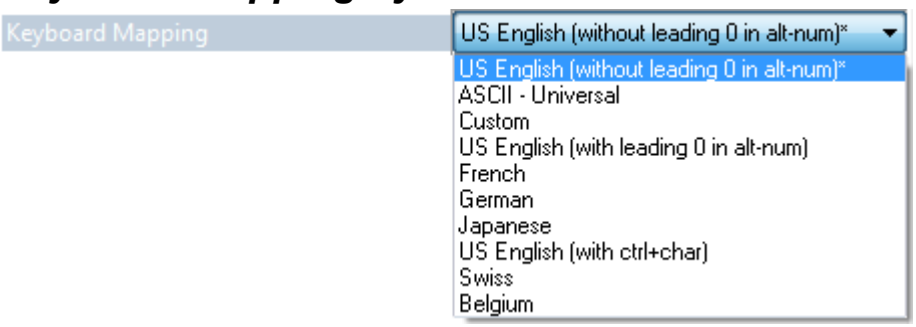

M0013_01
Universal


M0014_01
Custom


M0189_01
Belgian


M0190_01
Swiss

Keyboard Mapping by ESP



Communications Mode

Some **ESP** Communications options are unique to the software, and do not have corresponding programming symbols. These options are explained below.

Reader Packet Format

Data that is sent from the imager to the host in **Raw** format is sent without packet framing or check characters. **One-Way** communication is in a raw format, no response is expected from the host, and data is not resent.

Packetized data is sent with framing (a preamble communicating the amount of data to be transmitted, and a postamble containing error detection) and check characters, and a response is expected from the host. **Two-Way** communication is in packet format.

Reader to Host Packet Size

The **Reader to Host Packet Size** is the amount of data (in bytes) that is sent to the host in packet format. This feature allows you to set the maximum allowable packet size.

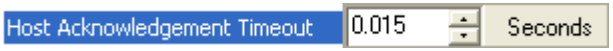
Expect Host Response

When **Expect Host Response** is enabled, the imager will re-transmit data if it doesn't receive acknowledgement from the host.

Reader Send Retry Count

Reader Send Retry Count sets the number of times the imager will re-transmit data before abandoning further send attempts. The minimum retry count is **1**, which represents the initial transmission.

Host Acknowledgement Timeout



The **Host Acknowledgement Timeout** is the amount of time (in seconds) that the imager will wait for an acknowledgement from the host before re-sending data.

Text Commands

When the **Text Commands** feature is enabled, the MS-2D can accept text commands via RS-232 connections and USB Virtual COM modes.

Note: Text Commands are not supported in USB HID Mode.

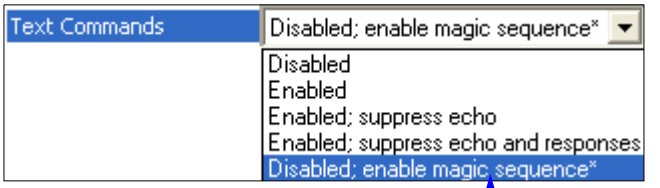
M0146_01

Enable Text Commands

M0147_01

Disable Text Commands (Default)

Text Commands by ESP



When **Magic Sequence** is enabled, it allows the user to enable **Text Commands** by entering a predetermined series of keystrokes.

When **Text Commands** are set to **Enabled; Suppress Echo**, text that a user enters in the terminal will not be shown. When **Text Commands** are set to **Enabled; Suppress Echo and Responses**, neither user-entered data or imager responses will be shown, and only decoded symbol data will appear in the terminal.

See **Terminal Right-Click Menu** for a way to change Echo settings directly in the terminal view.

Entering Magic Sequence

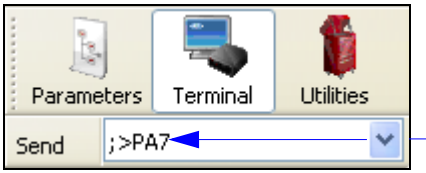
The magic sequence is **;>PA** followed by a numeric value of **1**, **3**, or **7**.

1 = Enable Text Commands

3 = Enabled; Suppress Echo

7 = Enabled; Suppress Echo and Responses

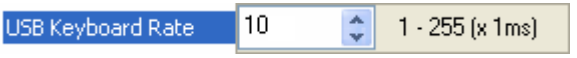
In the example below, the magic sequence entered will Enable Text Commands and Suppress Echo and Responses.



Enter the magic sequence in this text field and click **Send**.

Once the magic sequence has been sent, you can send text commands from the same text field.

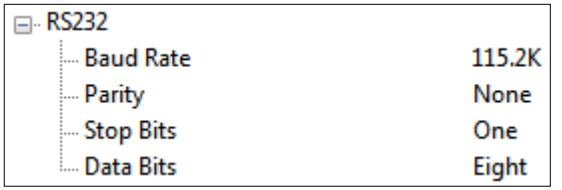
USB Keyboard Rate



Requests that the host polls the USB MS-2D at the rate specified (**1** to **255** ms).

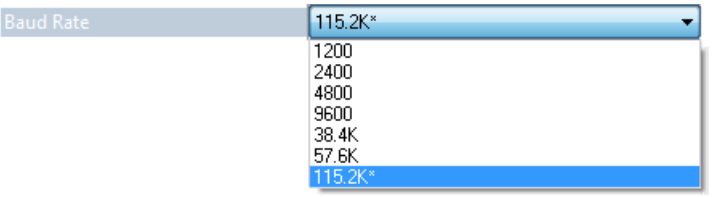
RS-232

See **RS-232 Interface** for RS-232 programming symbols.



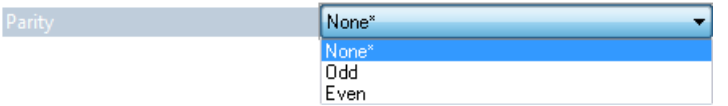
Baud Rate

Baud Rate is the rate at which the imager and host transfer data. It only needs to be changed if necessary to match the host setting.



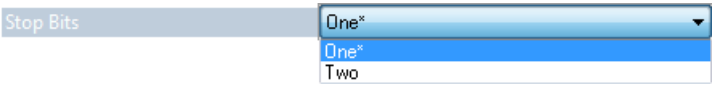
Parity

Parity is an error detection routine in which one data bit in each character is set to **1** or **0** so that the total number of 1 bits in the data field is even or odd. It only needs to be changed if necessary to match the host setting.



Stop Bits

Stop Bits are added to indicate the end of each character. This setting should only be changed if necessary to match the host setting.



Data Bits

Data Bits are the total number of bits in each character. This setting only needs to be changed if necessary to match the host setting.

Text Command Timeout

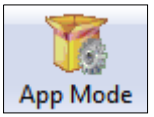
Text Command Timeout allows you to set the maximum time during which a complete text command from the host must be received. Pending text command data is discarded when the timeout is exceeded.

5 Read Cycle

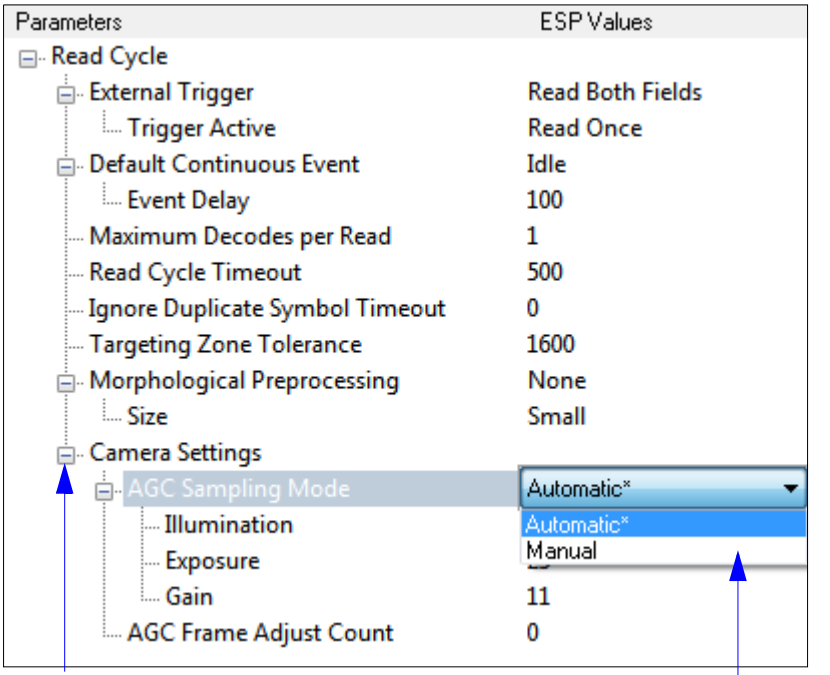
Contents

After you've established communications you will need to address the spatial and timing parameters associated with your application. This section explains those parameters. The Data Matrix symbols in this section can also be decoded to configure the MS-2D's Read Cycle parameters.

Read Cycle by ESP



Click this button to bring up the **App Mode** view, and then click the **Read Cycle** tab.

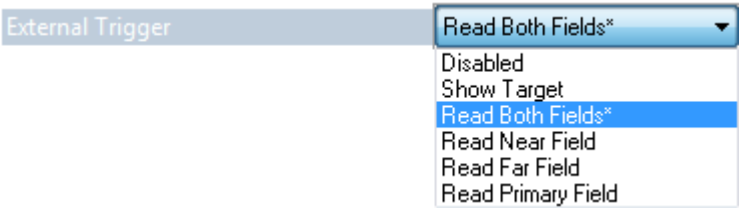


To open nested options, **single-click** the **+**.

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

External Trigger

The External Trigger parameter allows you to determine imager behavior when triggered externally.



Show Target

The target LEDs will illuminate when the imager is triggered externally.

Read Both Fields (Default)

Both Near Field and Far Field will be activated to capture an image when the imager is triggered externally.

Read Near Field

Near Field will be activated to capture an image when the imager is triggered externally.

Read Far Field

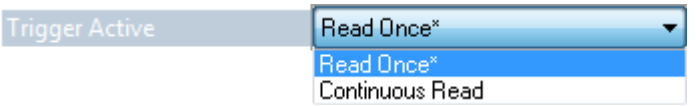
Far Field will be activated to capture an image when the imager is triggered externally.

Read Primary Field

When Read Primary Field is selected, the most recent field to have produced a Good Read (Near Field or Far Field) will be activated to capture an image when the imager is triggered externally.

Trigger Active

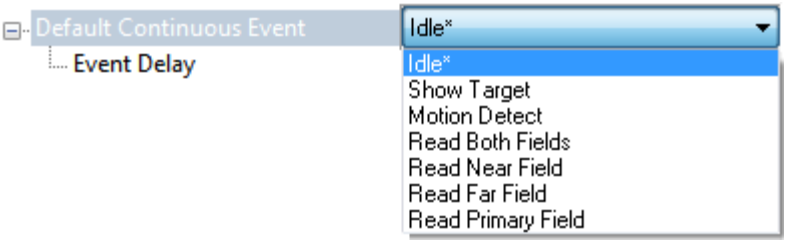
When an external trigger is active, the reader will either decode once and stop or decode continuously, depending on how this parameter is set. Trigger Active is set to Read Once by default.



Important: **Ignore Duplicate Symbol Timeout** should be set to a value greater than 0 when Trigger Active is set to Continuous Read.

Default Continuous Event

This parameter allows you to determine the default state of the imager.



Idle (Default)

When Default Continuous Event is set to Idle, the imager will remain inactive until triggered.

Show Target

When Default Continuous Event is set to Show Target, the imager will display the target LEDs but remain inactive until triggered externally.

Motion Detect

When Default Continuous Event is set to Motion Detect, the imager will remain inactive until motion occurs in the field of view (if a symbol is hand-presented, for example).

Read Both Fields

Both Near Field and Far Field will be continuously activated to capture an image.

Read Near Field

Near Field will be continuously activated to capture an image.

Read Far Field

Far Field will be continuously activated to capture an image.

Read Primary Field

When Read Primary Field is selected, the most recent field to have produced a Good Read (Near Field or Far Field) will be continuously activated to capture an image.

Event Delay

The default Event Delay is 0.100 seconds.

Maximum Decodes per Read

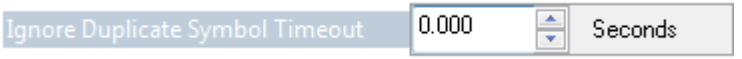
Maximum Decodes per Read allows you to set how many decodes can be performed in a single read cycle.

Read Cycle Timeout

Read Cycle Timeout determines the duration of the read cycle. The default Read Cycle Timeout is 0.500 seconds.

Ignore Duplicate Symbol Timeout

Ignore Duplicate Symbol Timeout sets the imager not to output the same symbol data multiple times within the time period designated.



Targeting Zone Tolerance

Targeting Zone Tolerance is particularly useful in environments where closely spaced symbols of various sizes need to be precisely targeted. It allows the imager to narrow the field of view relative to the size of a symbol, and to determine the distance the target must be from the symbol for a decode event to occur.

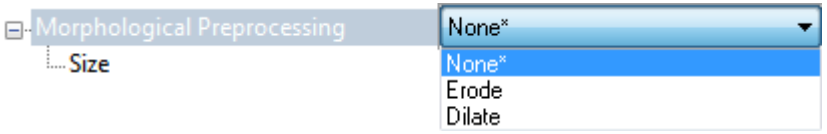
The default Targeting Zone Tolerance is 1600%.

Formula for calculating Targeting Zone Tolerance:

2 x distance from target to symbol (in pixels) / symbol width or height (in pixels) x 100

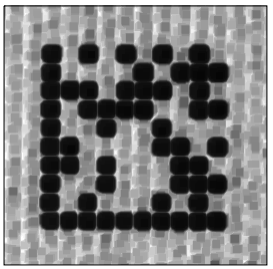
Morphological Preprocessing

Morphological Preprocessing allows you to select the method for processing captured images, and to choose the operator size for that method. It is set to None by default.



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.

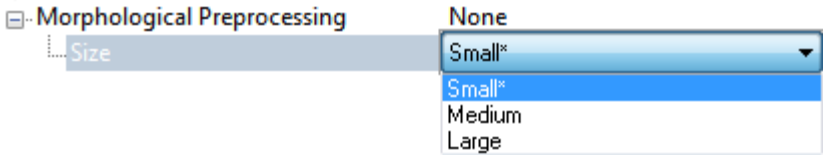


Dilate

Dilate increases the light cell size of a symbol. Useful for increasing the light cell size of a light-on-dark Data Matrix symbol.

Operator Size

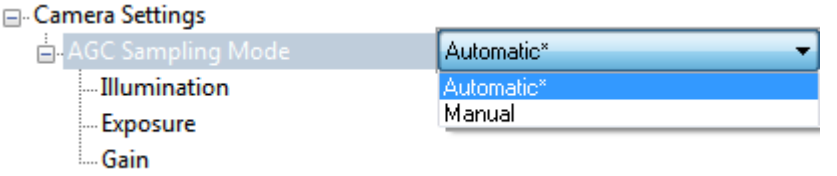
Operator Size determines the size of the area or “pixel neighborhood” (measured in pixels) in which the morphological operation is being performed.



Camera Settings

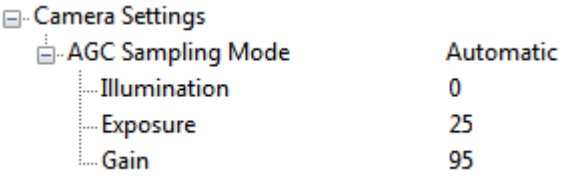
Camera Settings allow you to set AGC Sampling Mode and to set the percentage values for Illumination, Exposure, and Gain.

When AGC Sampling Mode is set to Automatic (default), each time a No Read occurs, the imager adjusts the gain and exposure for the next capture to optimize symbol contrast.



When AGC Sampling Mode is set to Manual, you are able to control the Illumination, Exposure, and Gain values yourself and the imager will use these parameters – which are now fixed – for each Read Cycle.

The values for Illumination, Exposure, and Gain can be set to any value between 0% and 100%. The default values are shown below.



AGC Frame Adjust Count

Automatic Gain Control (AGC) is a system that controls gain in order to maintain high performance over a range of input levels. Gain is essentially the ratio of output to input. Gain settings affect how the imager decodes symbols and captures images.

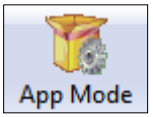
AGC Frame Adjust Count sets the number of image frames captured and discarded before the main image capture. This feature gives the gain control time to adjust.

6 Symbologies

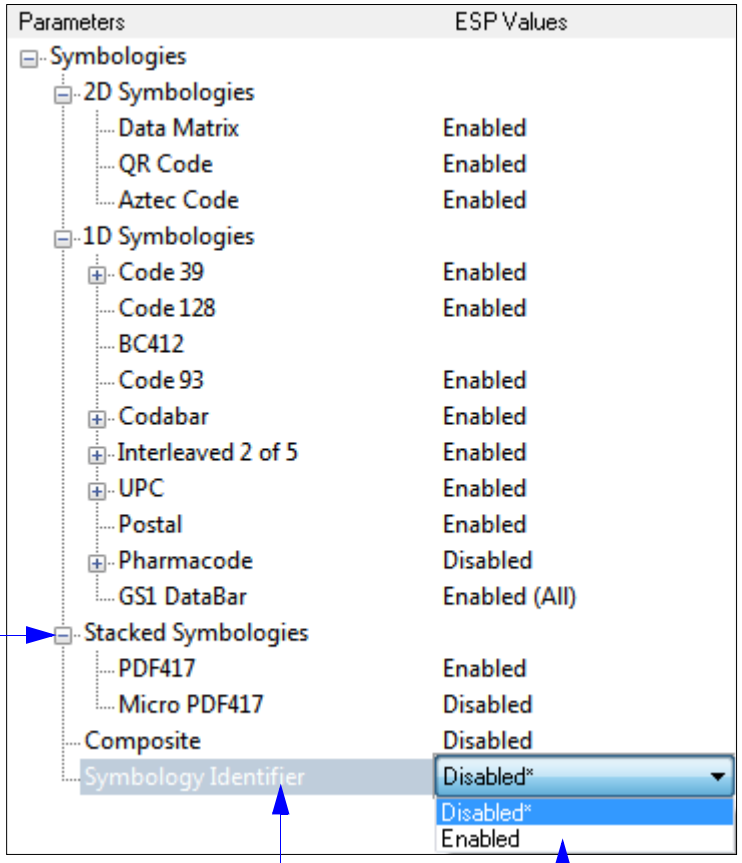
Contents

This section describes the various symbol types that can be decoded by the MS-2D Engine. The Data Matrix symbols in this section can also be decoded to configure the MS-2D's Symbologies parameters.

Symbologies by ESP



Click this button to bring up the **App Mode** view, and then click the **Symbologies** tab.



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

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

Data Matrix

Data Matrix	Enabled*
	Disabled
	Enabled*

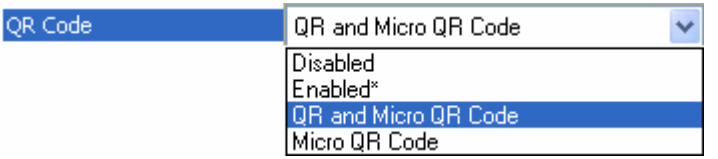
Sample Data Matrix Symbol

Note: The programming symbols for **Data Matrix Enabled** and **Data Matrix Disabled** are available on page A-11 in **Appendix C**.

If you disable the Data Matrix symbology, programming symbols will not be decodable by the MS-2D, and Data Matrix will need to be re-enabled using ESP.

Use the **Data Matrix Disabled** programming symbol with caution.

QR Code



QR Code Inverse and Standard Enabled

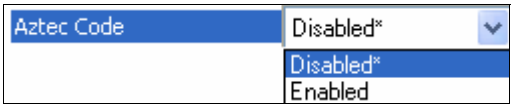
QR Code Disabled

QR Code and Micro QR Code Enabled

Sample QR Code Symbol

Sample Micro QR Code Symbol

Aztec



Aztec Enabled

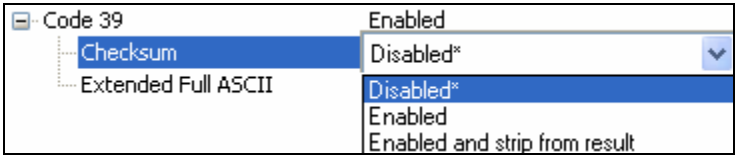

M0033_01

Aztec Disabled (Default)


M0034_01

Sample Aztec Symbol

Code 39



Code 39 Disabled


M0047_01

Code 39 Enabled (Default)


M0046_01

Code 39 Enable Checksum


M0048_01

Code 39 Disable Checksum (Default)


M0049_01

Code 39 Enable Checksum and Strip from Result


M0050_01

Code 39 Extended Full ASCII Enabled

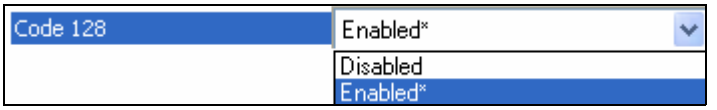

M0051_01

Code 39 Extended Full ASCII Disabled (Default)


M0052_01

Sample Code 39 Symbol

Code 128



Code 128 Enabled (Default)


M0061_01

Code 128 Disabled


M0062_01

Sample Code 128 Symbol


123456789A

BC412

BC412	Disabled*
	Disabled*
	Enabled

Sample BC412 Symbol

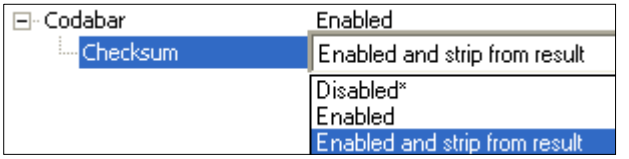
Code 93

Code 93 Enabled (Default)

Code 93 Disabled

Sample Code 93 Symbol

Codabar

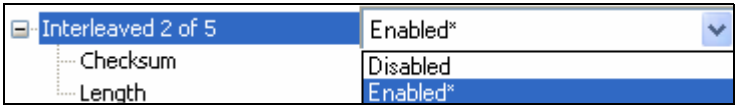


Codabar Enabled (Default)

Codabar Disabled

Sample Codabar Symbol

Interleaved 2 of 5



Interleaved 2 of 5 Enabled (Default)

Interleaved 2 of 5 Disabled

Interleaved 2 of 5 Two Digit Minimum

Interleaved 2 of 5 Four Digit Minimum

Interleaved 2 of 5 Six Digit Minimum

Sample Interleaved 2 of 5 Symbol

UPC

UPC Enabled (Default)

UPC Disabled

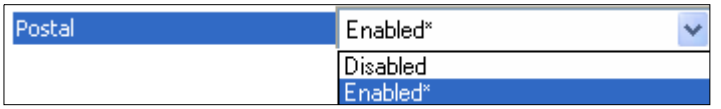
UPC Extension Enabled

UPC Extension Disabled

Sample UPC-E Symbol

Sample UPC-A Symbol

Postal



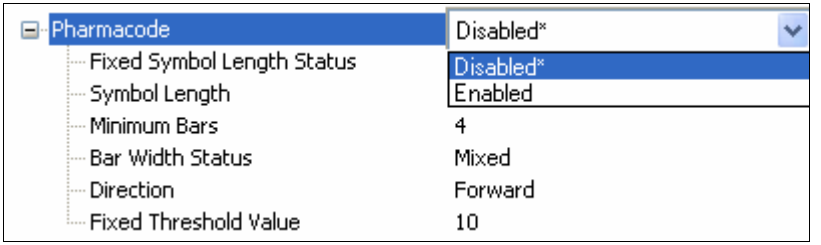
Supported Postal Symbologies

- Postnet
- Planet
- USPS4CB (Intelligent Mail)
- Australia Post
- Japan Post
- Royal Mail
- KIX
- UPU

Sample Postnet Symbol

Sample Royal Mail Symbol

Pharmacode



Fixed Symbol Length Status

When enabled, the imager will check the symbol length against the symbol length field. If disabled, any length will be considered valid.

Symbol Length

Specifies the exact number of bars that must be present for the imager to recognize and decode the Pharmacode symbol.

Minimum Bars

Sets the minimum number of bars that a Pharmacode symbol must have to be considered valid.

Bar Width Status

If set to **Mixed**, the imager 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 imager is able to tell the difference between the narrow and the wide bars.

Direction

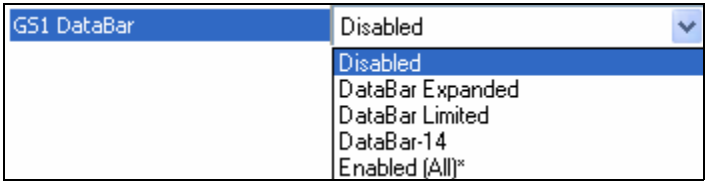
Specifies the direction in which a symbol can be read.

Fixed Threshold Value

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.

Sample Pharmacode Symbol

GS1 DataBar



All GS1 DataBar Enabled (Default)


M0116_01

All GS1 DataBar Disabled


M0117_01

GS1 DataBar Limited Enabled


M0112_01

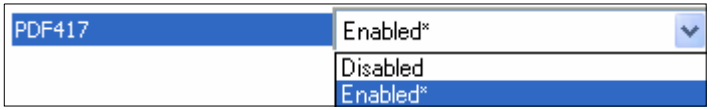
Sample DataBar-14 Limited Symbol

Sample DataBar-14 Stacked Symbol

Sample DataBar Expanded Symbol

Sample DataBar-14 Symbol

PDF417



PDF417 Enabled (Default)


M0091_01

PDF417 Disabled


M0092_01

Sample PDF417 Symbol

MicroPDF417

MicroPDF417 Disabled (Default)


M0084_01

MicroPDF417 Enabled


M0083_01

Sample MicroPDF417 Symbol

Composite

Composite consists of a 1D component associated with an adjacent 2D component. A successful decode is required for both the 1D and 2D components before the MS-2D outputs a result. When Composite is enabled, the unit decodes the 1D component first.

Important: EAN-8, EAN-13, UPC-A, and UPC-E cannot be decoded individually when Composite is enabled.

Maximum Decodes per Read

Maximum Decodes per Read represents the maximum number of candidate symbols in the field of view that can be decoded during a read cycle. Note that decode speed will decrease as the **Maximum Decodes per Read** value is increased.

Composite Disabled (Default)

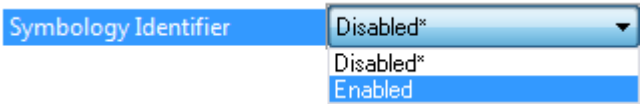
Composite Enabled

Sample Composite Symbol

Symbology Identifier

When **Symbology Identifier** is enabled, an AIM (Association for Automatic Identification and Mobility) preamble is added to decoded data output (see the **AIM Symbology Identifiers** list). This preamble identifies what kind of symbology has been decoded.

Symbology Identifier by ESP



AIM Symbology Identifiers

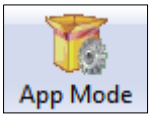
A	Code 39
C	Code 128
d	Data Matrix
e	GS1 DataBar / Composite
E	UPC/EAN
F	Codabar
G	Code 93
I	Interleaved 2 of 5
L	PDF417 / MicroPDF417
Q	QR Code / Micro QR Code
X	Other (Pharmacode)
z	Aztec

7 I/O Parameters

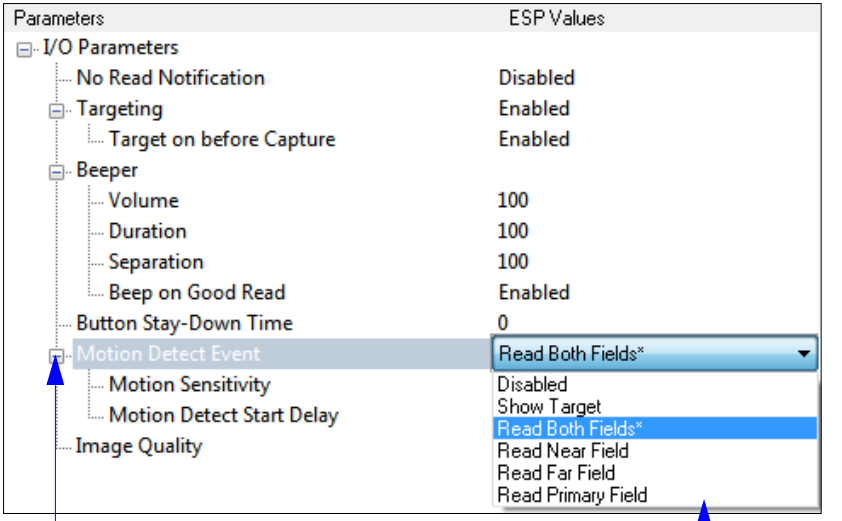
Contents

This section includes instructions on setting up conditions for changing input/output electrical transitions for control of the imager'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. The Data Matrix symbols in this section can also be decoded to configure the MS-2D's I/O parameters.

I/O Parameters by ESP



Click this button to bring up the **App Mode** view, and then click the **Read Cycle** tab.



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

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

No Read Notification

No Read Notification allows you to enable or disable user feedback alerting you when a symbol is not decoded successfully.

The No Read message output is **ap/r**, indicating that the imager failed to decode the symbol.

Targeting

The Targeting parameter allows you to turn the targeting LEDs on or off. They are on by default.

Read the configuration symbols below to enable or disable **Targeting**.


M0196_01

Targeting Off


M0197_01

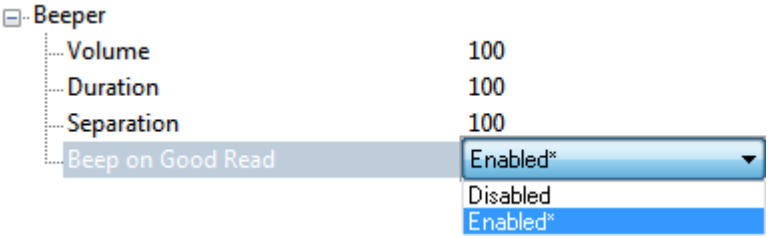
Targeting On

Beeper

The Beeper parameters allow you to set the Volume, Duration, and Separation of the beep, and whether or not it will beep on a Good Read.

Beeper volume is 100% by default, 0.100 seconds Duration by default, and 0.100 seconds Separation by default.

Beep on Good Read is enabled by default.



Read the configuration symbols below to enable or disable **Beeper**.

Beeper Off

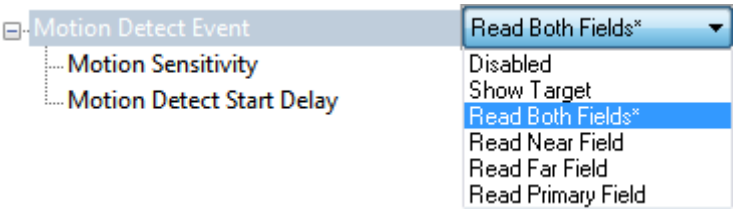
Beeper On

Button Stay-Down Time

Button Stay-Down Time sets the amount of time (in seconds) that the imager will continue to process the current “decode symbol” event. The imager will behave as if the trigger is being activated for this specified amount of time.

Motion Detect Event

Motion Detect Event allows you to determine the imager's behavior when motion is detected in the field of view.



Motion Sensitivity allows you to determine the sensitivity of motion detection (lower is more sensitive - 5 is default).

Motion Detect Start Delay allows you to set the amount of delay before a motion detect event occurs. (0 seconds is default.)

Show Target

The target LEDs will illuminate when a motion detect event occurs.

Read Both Fields (Default)

Both Near Field and Far Field will be activated to capture an image when a motion detect event occurs.

Read Near Field

Near Field will be activated to capture an image when a motion detect event occurs.

Read Far Field

Far Field will be activated to capture an image when a motion detect event occurs.

Read Primary Field

When Read Primary Field is selected, the most recent field to have produced a Good Read (Near Field or Far Field) will be activated to capture an image when a motion detect event occurs.

Read the configuration symbols below to enable or disable **Motion Detection**.


M0129_07

Motion Detection Off


M0162_01

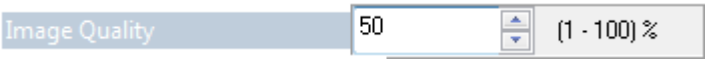
Motion Detection On, Start Delay 500 ms


M0161_01

Motion Detection On Start Delay 0

Image Quality

Image Quality allows you to determine the quality of images that are output from image captures. Image Quality is set to 50% by default.

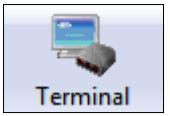


8 Terminal

Contents

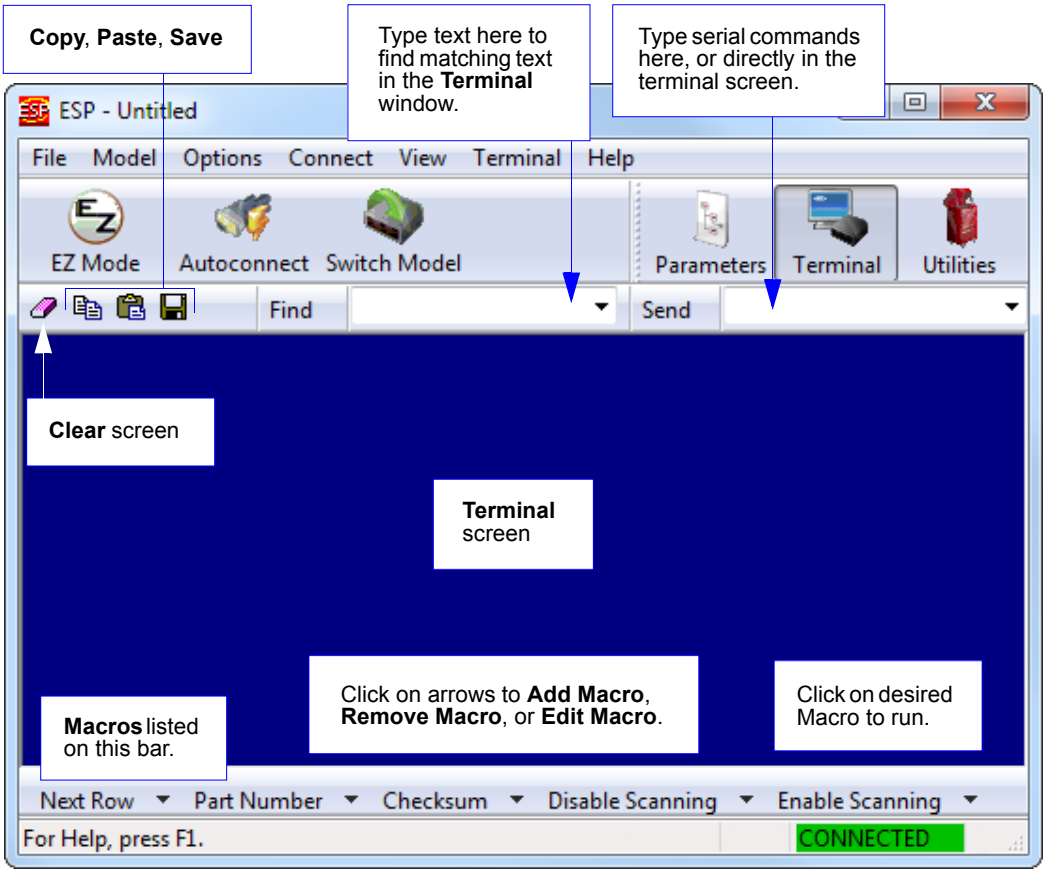
This section describes the **Terminal** window and **Macro** functions in **ESP**.

Terminal Window



Click this button to display the **Terminal** view.

The following view will appear:



The **Terminal** allows you to send serial commands to the MS-2D by typing commands in the **Send** text field or directly in the Terminal screen, by copying and pasting, or by using Macros.

The Terminal screen also displays symbol data or information from the MS-2D.

You can right-click on the screen to bring up a menu of several options for managing data in the Terminal view.

Find

The **Find** function allows you to enter text strings to be searched for in the Terminal window. For example, data from a series of symbols has been sent to the Terminal and you want to determine if a particular symbol whose data begins with “ABC” has been decoded.

1. Type “ABC” into the **Find** field.

2. Press **Enter**.

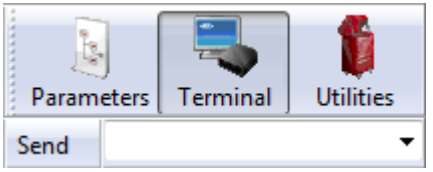
   The first instance of “ABC” will be highlighted in the Terminal window.

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 MS-2D.

1. Type the command into the **Send** box.

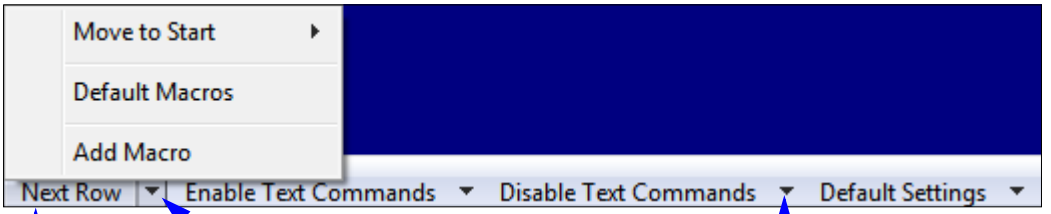


2. Press **Enter** to send the command to the MS-2D.
3. Click the **Send** button to the left of the text field to send the command again.

   See **Text Commands** in **Appendix D** for information on how to send Text Commands from Terminal.

Macros

Macros can be stored in the macro selection bar at the bottom of the Terminal view, edited in a separate window, and executed by clicking on the macro name.



Click on **Next Row** to see the next row of macros.

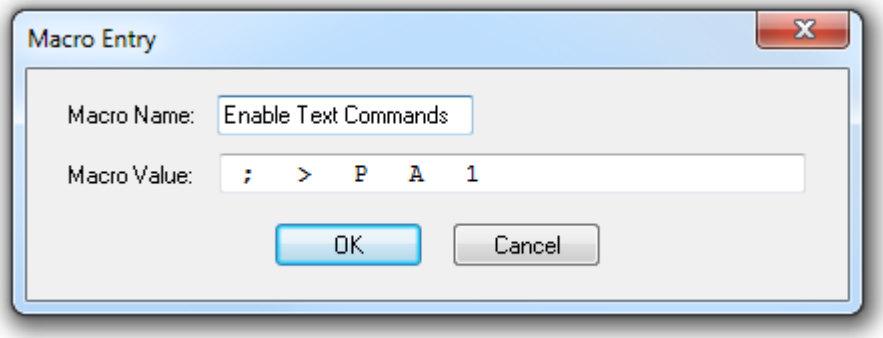
Click on the first arrow for **Move to Start** (moves any macro to the first position), **Default Macros** (restores all macros to original settings), and **Add Macro** (allows you to add your own macro).

Click on subsequent arrows for **Load to Current** (loads macro to current command), **Load to Send combo** (loads macro to command string), **Add Macro**, **Remove Macro** (removes current macro), and **Edit Macro** (allows you to edit current macro name or value).

When you click on the macro name, the macro is executed in the Terminal window. If the macro is a command, it is sent to the MS-2D at the same time that it is displayed.

Editing a Macro

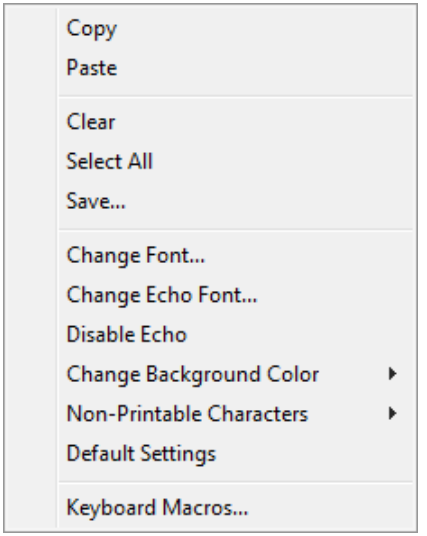
When you click the arrow next to any macro and select **Edit Macro**, the following dialog appears:



You can edit an existing **Macro Name** or **Macro Value** by typing a new name or value in the corresponding text field and clicking **OK**.

Terminal Window Menus

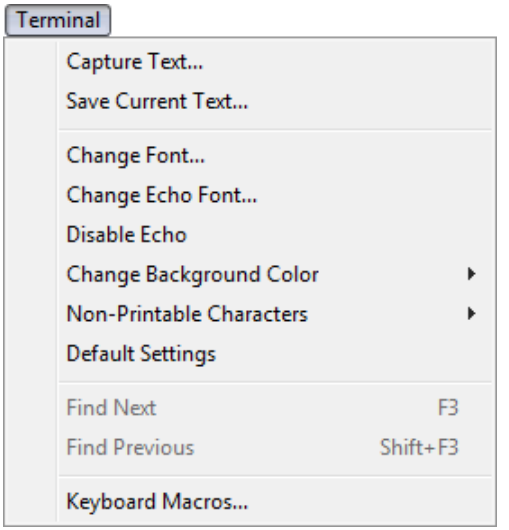
Right-click on the **Terminal** window to display the following menu:



- **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 of 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.



- **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.
- **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.

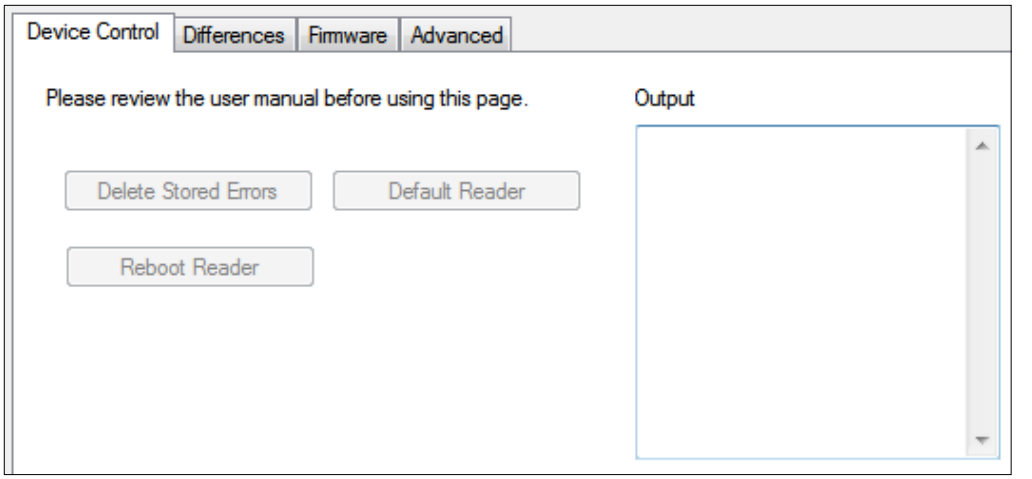
9 Utilities

Contents

This section explains ESP's Utilities features. These include Device Control, an interface that lets you perform delete, reboot, and default operations; Differences from Default, which shows all currently enabled settings that are not default settings; Firmware, where you can check firmware ID and other information about the imager; and Advanced, which allows you to collect batch files for customized imager configuration and optimization.

Device Control

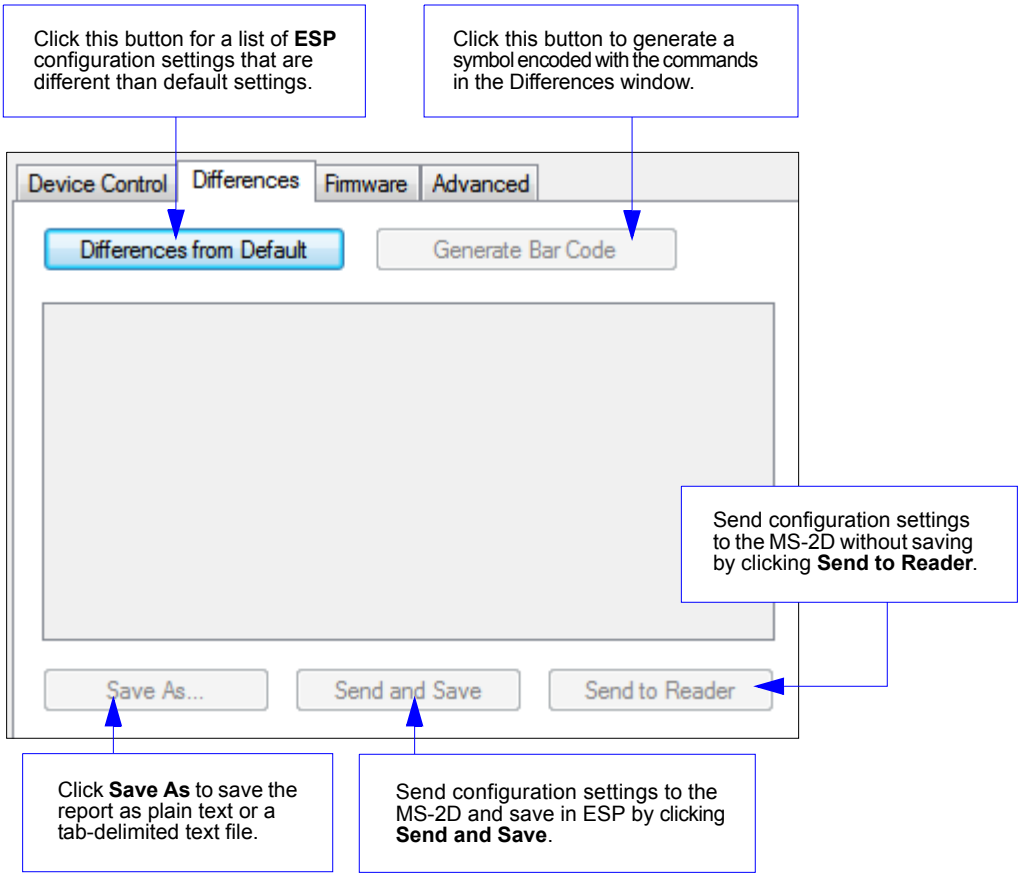
This feature allows you to delete stored errors, to reboot the imager, and to default the imager.



- **Delete Stored Errors** erases all logged errors whether you have looked at them or not.
- **Default Reader** returns the imager to its default state, without any optimization or configuration.
- **Reboot Reader** refreshes the imager's memory and functionality, returning it to the most recent configuration you have saved.

Differences from Default

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.

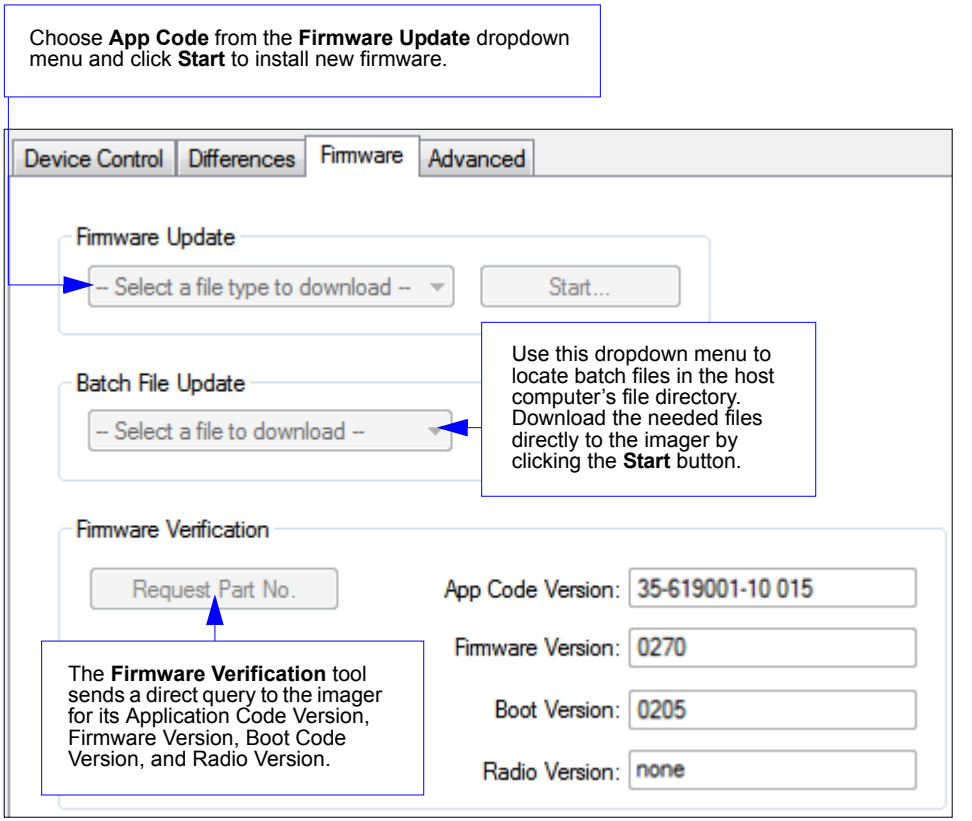


- To save the **Differences from Default** report, either as plain text or as a tab-delimited text file, click **Save As**.
- Click **Send and Save** to send the settings to the MS-2D and save them, or **Send to Reader** to send the settings without saving them.

Important: The use the **Differences from Default** feature, you must connect to the MS-2D and **Receive Reader Settings** via the **Send/Recv** button on the toolbar.

Firmware

The Firmware view in ESP Utilities is a simple way to update and verify your imager's firmware and to update batch files.



ID and Firmware Version

Another way to query the 2D Engine for its identifying information is by reading the following symbol:


M0148_01

ID and Firmware Version

The host's text program will output a data string containing the device's identifying information in the format shown below.

Example:

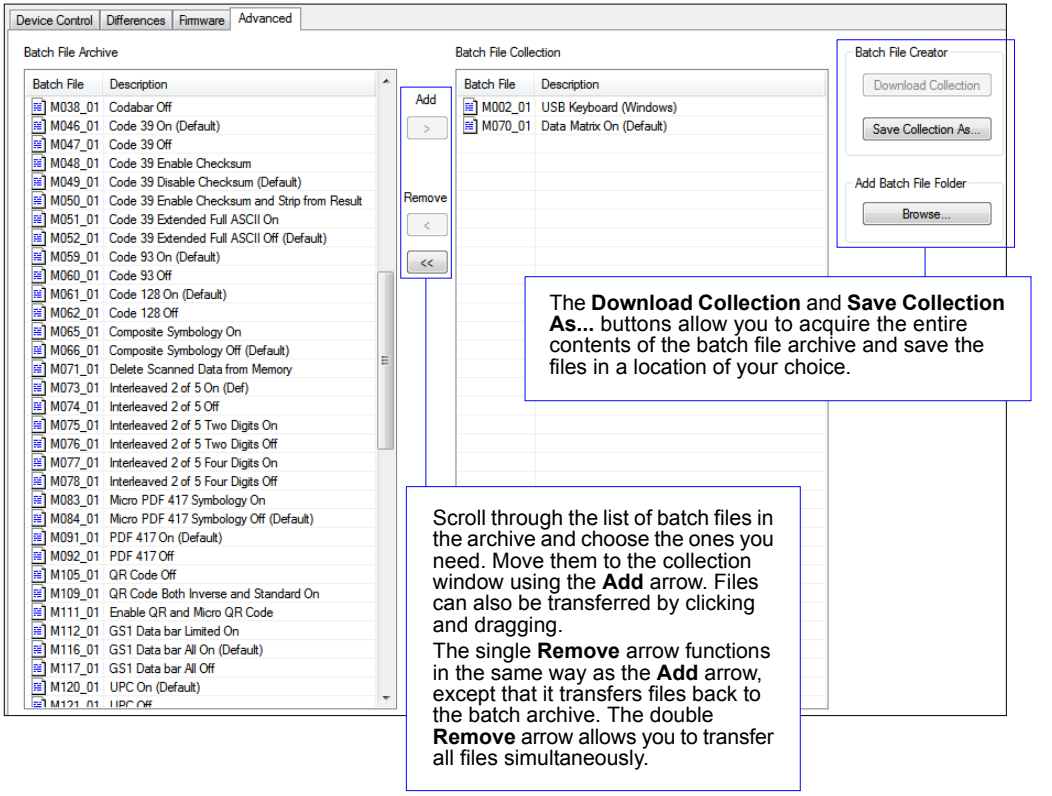
i02700205none0020008143A06000000600060013004800 02<TAB>35-619001-10 015

0270	Application Version Number
0205	Bootloader Firmware Version
None	Radio Firmware Version
0020008143	Serial Number
A	A – Running Application
06	N/A
0	N/A
0000	N/A
06	Hardware Identifier
0006	Hardware Type Identifier – 0006 is MS-2D
0013	Boot Application Version
0048	Operating System Kernel Version
0002	Root File System Versions
<TAB>	ASCII TAB Character
35-619001-10 015	Decoder Version PN and BN

Advanced

The **Advanced** tab in **Utilities** features an archive of all batch files containing imager configuration commands. Each batch file's extension is .crb, and each file contains the fundamental code for programming the imager. Notice that the names of the batch files correspond with the numbers beneath all the Data Matrix configuration symbols.

This tool allows you to use the batch file data to create your own symbols, or to collect only the files that you use frequently to configure the imager for your application.



Appendices

Contents

Appendix A — General Specifications

Mechanical (MS-2D Engine Only)
Height: 0.47” (11.9 mm)
Width: 0.81” (20.6 mm)
Length: 0.57” (14.5 mm)
Weight (Without Mounting Tabs): 0.09 oz. (2.6 g)

Mechanical (MS-2D with Decode Board)
Height: 1.54” (39.0 mm)
Width: 0.98” (24.9 mm)
Length: 0.30” (7.6 mm)
Weight: 0.10 oz. (2.8 g)

Mechanical (Integrated Unit)
Height: 1.54” (39.0 mm)
Width: 0.98” (24.9 mm)
Length: 0.86” (21.8 mm)
Weight: 0.91 oz. (25.7 g)

Light Source
Type: High-output red LED illumination
Target: Blue LED

Light Collection Options
Sensor: CMOS 1.2 megapixel
Optical Resolution: 960 x 640 pixels, high density and wide density

Scanning Performance
Pitch: ± 60° (from front to back)
Skew: ± 60° from plane parallel to symbol (side-to-side)
Tilt: ± 180°
Print Contrast Resolution: 25% (1D symbologies), 35% (2D symbologies) absolute dark/light reflectance differential, measured at 650 nm

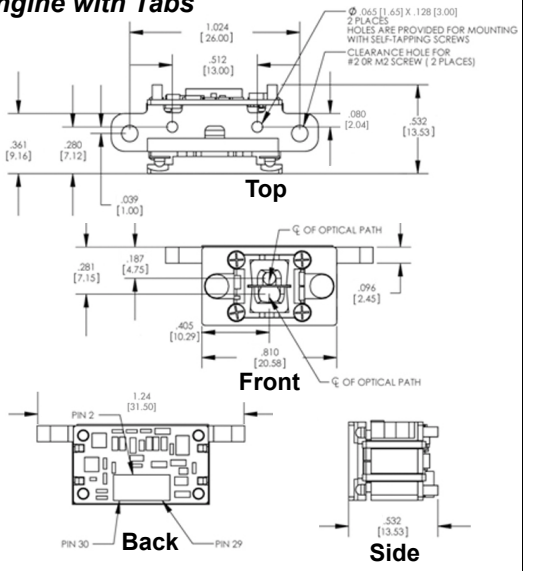
Communication
Interface: USB 2.0, RS-232 TTL
Memory: 128 MB Flash ROM, 32 MB RAM

Symbologies
2D Symbologies: Data Matrix (ECC 0-200), QR Code, Micro QR Code, Aztec
Stacked Symbologies: PDF417, MicroPDF417, GS1 DataBar (Composite and Stacked)
1D Symbologies: Code 39, Code 128, BC412, Interleaved 2 of 5, UPC/EAN, Codabar, Code 93, Pharmacode, Planet, Postnet, Japan Post, Australian Post, Royal Mail, Intelligent Mail, KIX

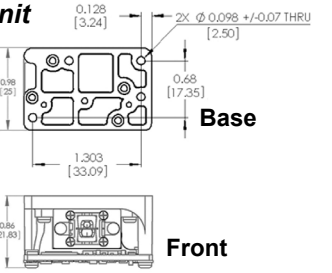
Environmental
Operating Tempurature: –20° to 55° C (–4° to 131° F) **Storage Tempurature:** –30° to 65° C (–22° to 150° F)
Relative Humidity: 5% to 95% (non-condensing)
Ambient Light Immunity: Sunlight: Up to 9,000 ft.-candles /96,890 lux
Shock: Withstands multiple drops of 6’ (1.8 meters) to concrete in an enclosed housing

Indicators
Symbol Locator: Blue LED targeting beam
Beeper: Good Read

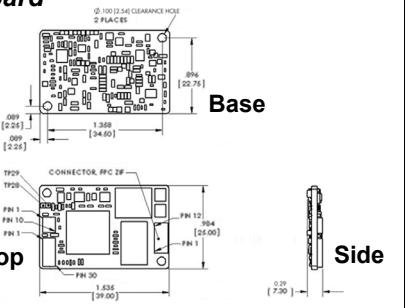
Engine with Tabs



Integrated Unit



Decode Board



Note: Nominal dimensions shown. Typical tolerances apply.

MS-2D Engine Dimensions

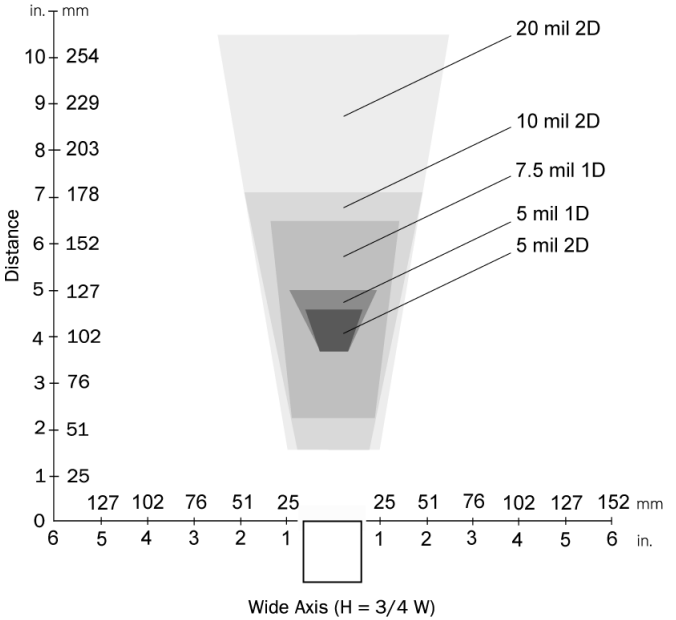
Electrical

Power: 5VDC
Typical: 303 mA **Idle:** 57 mA **Sleep:** 1.6 mA

CE Standards

Emissions: EN 55022:2006 Class B Limits; ICES-003
LED Safety: IEC 62471:2006 (Ed. 1.0)

Read Ranges



Wide Axis (H = 3/4 W)

Narrow Bar	Read Range
STANDARD DENSITY	
1D	
.0050” (.127 mm)	3.7 to 5.0” (94 to 127 mm)
.0075” (.191 mm)	2.2 to 6.5” (56 to 165 mm)
.010” (.254 mm)	1.5 to 8.0” (38 to 203 mm)
.020 (.508 mm)	2.3 to 15.5” (58 to 394 mm)
2D	
.0050” (.127 mm)	3.7 to 4.6” (94 to 117 mm)
.0075” (.191 mm)	1.5 to 6.0” (38 to 152 mm)
.010” (.254 mm)	1.6 to 7.1” (41 to 180 mm)
.020 (.508 mm)	1.6 to 10.5” (41 to 267 mm)

FIS and Demo Kit Options

MS-2D Engine	
MS-2D Engine, USB, Tabs, 12 in. Ribbon, Standard Focus	FIS-MS2D-0001G
MS-2D Engine, USB, Bracket, 12 in. Ribbon, Standard Focus	FIS-MS2D-0002G
MS-2D Engine, Serial, Tabs, 12 in. Ribbon, Standard Focus	FIS-MS2D-0003G
MS-2D Engine, Serial, Bracket, 12 in. Ribbon, Standard Focus	FIS-MS2D-0004G
Demo Kit, MS-2D, USB Engine, Decode, Bracket, Interface Board, 6 in. Ribbon	98-000224-01
Demo Kit, MS-2D, Serial Engine, Decode, Bracket, Interface Board, 6 in. Ribbon	98-000224-02

Safety Certifications

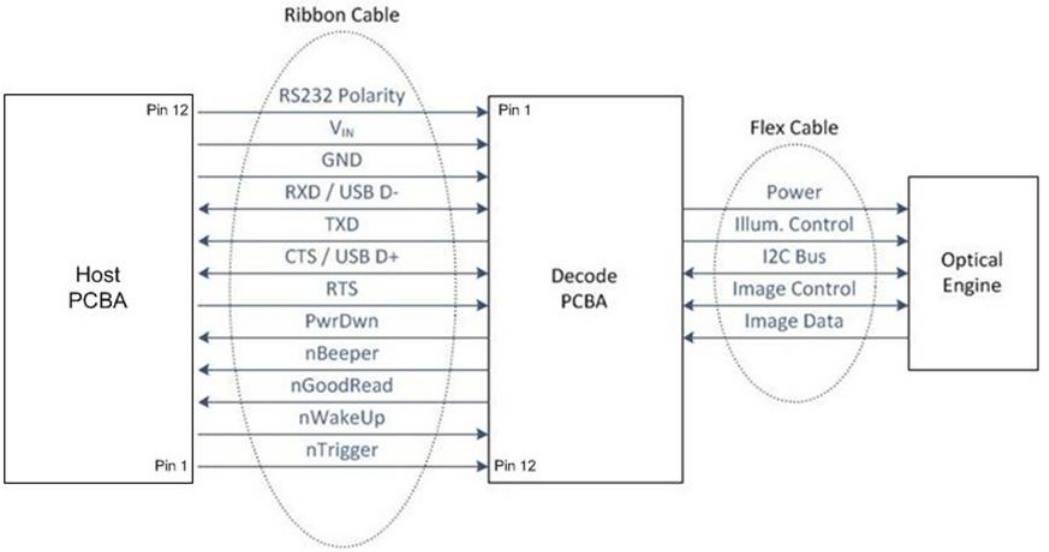
FCC, CE, RoHS/WEEE

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. Standard Warranty–One-Year Limited Warranty on parts and labor. Extended 3-Year Warranty available.

Appendix B — Electrical Specifications

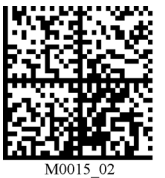
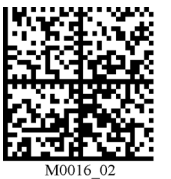
Power: 5VDC (mA)
Typical: 303 mA **Idle:** 57 mA **Sleep:** 1.6 mA

Electrical System



Note: The part number for the ribbon cable FPC ZIF connector on the decode board is FCI 10051922-1210ELF. See “Bottom View” in **Decode PCB Mechanical Specifications** for the location of the ribbon cable connector.

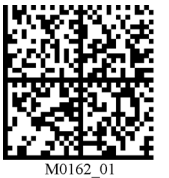
Appendix C — Configuration Symbols

A1  M0002_01 USB Keyboard (Windows)	**A2**  Factory Default Default to Factory Settings	**A3**  M0004_01 USB Native (HID) Mode	**A4**  M0005_01 USB Virtual COM Mode
	B2  M0007_02 U.S. Keyboard Mapping (Default)	**B3**  M0008_02 U.S. Keyboard without Leading 0	**B4**  M0009_02 U.S. Keyboard with Ctr+Char
C1  M0010_01 French Keyboard Mapping	**C2**  M0011_01 German Keyboard Mapping	**C3**  M0012_01 Japanese Keyboard Mapping	**C4**  M0013_01 Universal Keyboard Mapping
D1  M0014_01 Custom Keyboard	**D2**  M0015_02 USB Enable Alternate OS	**D3**  M0016_02 USB Disable Alternate OS	**D4**  M0018_01 RS-232 7 Data Bits
E1  M0019_01 RS-232 8 Data Bits (Default)	**E2**  M0020_01 RS-232 1200 Baud	**E3**  M0021_01 RS-232 2400 Baud	**E4**  M0022_01 RS-232 4800 Baud

A1  M0023_01 RS-232 9600 Baud	**A2**  M0024_01 RS-232 19200 Baud	**A3**  M0025_01 RS-232 38400 Baud	**A4**  M0026_01 RS-232 57600 Baud
B1  M0027_01 RS-232 115200 Baud (Default)	**B2**  M0028_01 RS-232 Even Parity	**B3**  M0029_01 RS-232 Odd Parity	**B4**  M0030_01 RS-232 No Parity (Default)
C1  M0031_01 UART Flow Control None (Default)	**C2**  M0032_01 UART Flow Control Hardware	**C3**  M0033_01 Aztec On	**C4**  M0034_01 Aztec Off (Default)
D1  M0037_01 Codabar On (Default)	**D2**  M0038_01 Codabar Off	**D3**  M0046_01 Code 39 On (Default)	**D4**  M0047_01 Code 39 Off
E1  M0048_01 Code 39 Enable Checksum	**E2**  M0049_01 Code 39 Disable Checksum (Default)	**E3**  M0050_01 Code 39 Enable Checksum and Strip from Result	**E4**  M0051_01 Code 39 Extended Full ASCII On

A1  M0052_01 Code 39 Extended Full ASCII Off (Default)	**A2**  M0059_01 Code 93 On (Default)	**A3**  M0060_01 Code 93 Off	**A4**  M0061_01 Code 128 On (Default)
B1  M0062_01 Code 128 Off	**B2**  M0065_01 Composite On	**B3**  M0066_01 Composite Off (Default)	
C1  M0071_01 Delete Scanned Data from Memory	**C2**  M0073_01 Interleaved 2 of 5 On (Default)	**C3**  M0074_01 Interleaved 2 of 5 Off	**C4**  Interleaved 2 of 5 Two Digit Minimum
D1  Interleaved 2 of 5 Four Digit Minimum	**D2**  Interleaved 2 of 5 Six Digit Minimum		**D4**  M0083_01 MicroPDF417 On
E1  M0084_01 MicroPDF417 Off (Default)	**E2**  M0091_01 PDF417 On (Default)	**E3**  M0092_01 PDF417 Off	**E4**  M0105_01 QR Code Off

A1  M0109_01 QR Code Inverse and Standard On	**A2**  M0111_01 Enable QR Code and Micro QR Code	**A3**  M0112_01 GS1 DataBar Limited On	**A4**  M0116_01 All GS1 DataBar On (Default)
B1  M0117_01 All GS1 DataBar Off	**B2**  M0120_01 UPC On (Default)	**B3**  M0121_01 UPC Off	**B4**  M0124_01 UPC Extension On
C1  M0125_01 UPC Extension Off	**C2**  M0126_03 Continuous Trigger Off	**C3**  M0129_07 Motion Detection Off	**C4**  M0130_02 Preamble - Comma
D1  M0131_02 Preamble - Space	**D2**  M0132_02 Preamble - Tab (Keyboard Only)	**D3**  M0133_02 Preamble - Tab (Serial Only)	**D4**  M0134_02 Preamble - Erase (None)
E1  M0135_02 Preamble - CR LF (Serial Only)	**E2**  M0136_02 Postamble - CR (Serial Only)	**E3**  M0137_02 Postamble - Comma	**E4**  M0138_02 Postamble - LF (Serial Only)

A1  M0139_02 Postamble - CR LF (Serial Only)	**A2**  M0140_02 Postamble - Space	**A3**  M0141_02 Postamble - Enter (Keyboard Only)	**A4**  M0142_02 Postamble - Tab (Keyboard Only)
B1  M0143_02 Postamble - Tab (Serial Only)	**B2**  M0144_02 Postamble - Erase (None)	**B3**  M0145_02 Preamble and Postamble - Erase	**B4**  M0146_01 Reader Text Commands On
C1  M0147_01 Reader Text Commands Off	**C2**  M0148_01 Output Information String	**C3**  M0149_01 RS-232 Raw Mode (Default)	**C4**  M0150_01 RS-232 Packet Mode
D1  M0151_01 Interleaved 2 of 5 with Control Character Stripped	**D2**  M0161_01 Motion Detect On, Start Delay 0	**D3**  M0162_01 Motion Detect On, Start Delay 500 ms	**D4**  M0165_01 Clear All JavaScript Rules
E1  M0166_01 Save Settings	**E2**  M0167_01 Beep Off	**E3**  M0168_01 Beep On	**E4**  M0189_01 Belgian Keyboard Mapping

A1  M0190_01 Swiss Keyboard Mapping	**A2**  M0191_01 Disable Duplicate Symbol Timeout	**A3**  M0192_01 1 Second Duplicate Scan Delay	**A4**  M0193_01 2 Second Duplicate Scan Delay
B1  M0194_01 3 Second Duplicate Scan Delay	**B2**  M0195_01 Default Trigger Delay	**B3**  M0196_01 Targeting Off	**B4**  M0197_01 Targeting On

C1  Data Matrix On	**C2**  Data Matrix Off

Important: If you disable the Data Matrix symbology, programming symbols will not be decodable by the MS-2D, and Data Matrix will need to be re-enabled using ESP.

Use the **Data Matrix Disabled** programming symbol with caution.

Configuration Symbol Reference

Appendix D — Serial Commands

Text Commands

Text commands may be sent to the reader in RS-232 or USB Virtual COM mode using any serial communications software, e.g., ESP Terminal.

Encoded-data is decoded by the reader by replacing %xx by a single byte with the value specified by the two hex-digits xx, e.g., **%25** would be replaced by character number 0x25, which is ASCII '%'.

text-command: *command-type encoded-dataopt carriage-return*

command-type: Single ASCII character in the set defined in **Command Types**.

encoded-data: *encoded-datum I encoded-data encoded-datum*

encoded-datum: *printable-character | % hex-digit hex-digit*

printable-character: any byte value in the range [0x20,0x7e]

hex-digit: **'0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9' |'A' | 'B' | 'C' | 'D' | 'E' | 'F' | 'a' | 'b' | 'c' | 'd' | 'e' | 'f'**

carriage-return: **0x0d**

In order to eliminate inadvertent commanding of the reader, Text Commands are disabled by default. To enable Text Commands requires an initial sequence: ;>PAx where x is as defined in the Reader Settings Table, register setting 41. (Note: 'A' is the ASCII character that corresponds to 41 HEX.)

For example, to send the reader commands by typing commands in ESP Terminal:

;>PA1

P(xx)yy

P(xx)yy

~

PA8

Where **;>PA1** enables text commands with echo and command responses; **P%xxyy** can be any desired commands; **~** saves the settings just sent (the ~ command saves all but communication-related settings); and **PA8** turns text commands back off (except for the initial sequence). (**Note:** *'A'* is the ASCII character that corresponds to 41 hex, thus P%418 would be equivalent.)

Note: ;>PA1 is used for interactive text commands. If the commands are to be saved in a file and sent non-interactively, use **;>PA7** instead; this enables text commands but disables echo and command responses. (See **Command Types**, **Reader Settings**, and **CRB System** for additional information.)

The following two examples can be sent to a reader in RS-232 mode from ESP Terminal by just typing the example text.

Example 1 (make the reader beep 3 times):
#%03 *Expected output: should make reader beep 3 times*

Example 2 (set reader to continuous read, High Density field (FOI0) only):
P(C4)5 *Expected output: should set reader to continuous read, High Density field (FOI0) only*

Example 3 (set reader to trigger read mode):
P(C4)255 *Expected output: should set reader to trigger read*

Example 4 (trigger the reader):
$%03 *Expected output: should trigger the reader*

Packetized Commands

Packetized commands consist of packetized data sent from Host-to-Reader to configure and cause the reader to perform certain functionalities (e.g. settings). Packetized commands are always enabled, unlike text commands. In addition, they include error detection data, making them more robust than text commands.

normal-command structure:

[preamble] [command-type] [data-size] [dataopt] [reserved field] [crc14]

preamble:	**0xEE 0xEE 0xEE 0xEE**
command-type:	Single ASCII character in the set defined in **Command Types**.
data-size:	byte value in range [0,240], which indicates size of data (in bytes) following this before *[reserved field]*
dataopt:	*datum* or *data datum* *data* or *datum* is any byte value in the range [0,255]
reserved field:	**0x00**
crc14:	Two consecutive bytes, each in range [0,127], representing the crc16 value and with the value 0x7F7F, most significant byte first. The packet crc16 is calculated over the entire packet, excluding the preamble and the crc14 itself. (See source files *crc16.[hc]* (Appendix) for details on the crc16 algorithm and polynomials to be used.)

Note: Most terminal programs will omit or not transmit a NULL (0x00) character entered in the terminal. So, if one tries to compose the command structure manually and send it through a terminal program, one might have to send everything before the reserved field, then send the 0x00 (key Ctrl+@ in ESP terminal view) by itself, and then the CRC14 checksum bytes.

In the first of the examples shown below, for instance, “0xEE 0xEE 0xEE 0xEE 0x23 0x01 0x03” can be sent using ESP’s Terminal, then the NULL character can be sent by typing “Ctrl + Shift + 2”, and then “0x4E 0x71” can be sent to complete the command sequence.

Examples:

0xEE 0xEE 0xEE 0xEE 0x23 0x01 0x03 **0x00** 0x4E 0x71

This executes the beep command, causing the reader to beep 3 times. "#%03" in text command format. The first 4 0xEE are the preamble, followed by the '#' character or 23h the beep command, and followed by data size of 1 with a data datum of 3 following it. 0x00 is the reserved field followed by two bytes CRC14 of the command + data size + data datum.

0xEE 0xEE 0xEE 0xEE 0x49 0x00 **0x00** 0x03 0x3C

This executes the info command, "I" in text command format. Since it does not have any data datum associated with it, 0x00 following 0x49 (ASCII "I") but before the reserved field and CRC14 (0x00 0x03 0x3C) indicating it has no *dataopt*.

0xEE 0xEE 0xEE 0xEE 0x50 0x04 0xC4 0x32 0x35 0x35 **0x00** 0x43 0x3C

This will change the reader to trigger-read mode, "P(C4)255" in text command format. 0x50 is the command-type, 0x04 indicates that following 4 bytes are the data option with 0xC4 being the register that needs to be updated and 0x32 0x32 0x35 being the new value.

0xEE 0xEE 0xEE 0xEE 0x50 0x03 0xA9 0x35 0x63 **0x00** 0x77 0x6B

This will change the reader Bypass Gain to 92 (0x5c) percent, "P(A9)5c" in text command format. 0x50 is the command-type, 0x03 indicates that the following 3 bytes are the data option with 0xA9 being the register that needs to be updated and 0x35 0x63 (5c) being the new value.

0xEE 0xEE 0xEE 0xEE 0x50 0x07 0x28 0x31 0x45 0x41 0x29 0x46 0x41 **0x00** 0x14 0x18

This will change Good Read Duration to 250ms, "P(1EA)FA" in text command format. In this case the register 1EA is more than one byte can hold; the register is converted to its individual ASCII hex value bounded by 0x28 and 0x29, underlined above.

0xEE 0xEE 0xEE 0xEE 0x24 0x01 0x03 **0x00** 0x1F 0x5C

This will trigger the reader, "$%03" in text command format. The reader will respond with: 0x01 0x58 0x1e 0x61 0x70 0x2f 0x64 0x04 indicating that it has accepted the trigger command. If a symbol is decoded, the decoded symbol data will follow the response. Example: The reader will respond to the trigger command with: {SOH}X{RS}ap/d{EOT}123test where "123test" is the symbol data the reader has decoded.

Command Types

#	Causes the reader to beep the specified number of times; *data* contains the number as a single character in the range [0,127]. (The reader will respond with *d* or *e*.) Example – beep three times: #%03
$	Posts an event to the reader; *data* contains the event number as a single character. See setting 39 in **Reader Settings** for a list of the event numbers. (The reader will respond with *d* or *e*.)
(	Causes the reader to upload any logged error messages (no *data*) (The reader will respond with a g packet, zero or more z packets, and a final *d* or *e*. Each *z* packet contains a portion of the requested data in its *data* field. **Note:** This is very similar to the response to the X command; however, *p* packets are not applicable and the *g* and *d/e* packets are not suppressed even in raw mode.)
)	Causes the reader to erase its log of error messages (no *data*) (The reader will respond with *d* or *e*.)
,	Causes the reader to send a list of current reader settings (no *data*) (The reader will respond with d containing a space-separated list of all setting values (in order, expressed as hexadecimal ASCII characters) or with *e*.)
/	Toggle a bit (or bits) in a reader setting; *data* contains a printable ASCII string in the following format: hexadecimal setting number in parentheses followed by a 32-bit signed integer value, expressed in ASCII hexadecimal characters (with optional minus sign) or ASCII decimal characters preceded by the '#' character, e.g., /(2e)1000 or /(2e)#4096; the specified integer is XOR'ed with the existing setting value. (The reader will respond with *d* or *e*.) **Note:** See **Reader Settings** for possible reader settings.
1	Indicates the start of a file download; *data* is empty. This command is followed by a sequence of *2* commands containing the file data and a download-end command (e.g., *5*). (The reader will respond with *d* or *e*.)
2	Indicates a continuation of a file download; *data* contains the next portion of the file data. (The reader will not send any response.)
5	Indicates the end of a regular file download; *data* contains the name of the file, which is from 1 to 200 letters, digits, periods, hyphens, and underscores, terminated with ASCII NUL. (The reader will respond with *d*, *e*, or *f*.)

9	Requests the reader to delete a file from its storage; *data* contains the file name, terminated with ASCII NUL.
;	Reserved (nop – treated as a comment)
<	Causes the reader to send a list of saved reader settings (no *data*) (see ',' command)
=	Puts setting directly to reader's non-volatile memory so that it will take effect upon next reboot; *data* is as defined in the / command; the specified integer replaces the existing setting value. **Note:** This command can be used to set communication modes without losing communication during the process. (The reader will respond with *d* or *e*.)
>	Causes the reader to send a string of text to the host as a *z* packet; *data* contains the text to send. (The reader will respond with a z packet containing the text.)
@	Causes the reader to reset its internal date/timestamp to the specified time; *data* contains the date and/or time in one of the following formats. yyyy-mm-dd hh:mm:ss yyyy-mm-dd hh:mm hh:mm:ss hh:mm **Note:** The separators are optional; only digits are significant. (The reader will respond with *d* or *e*.) Examples: Set to midnight: @00:00 Set to Sept 1, 2005 11:52:02 PM: @2005-09-01 23:52:02 **Note:** On units without a battery-backed real-time clock, the date and time will reset to 2000-01-01 00:00:00 upon power-up.)
A	Notifies the reader that the previously sent data were rejected for one of the following reasons: • The packet was encrypted and the decryption failed. • The host is locked to a different reader. The reader should indicate to the user that the packet has been rejected; e.g., it may sound error beeps. See related setting 0x12f, notify-of-packet-rejection. (The reader will not respond to the host.)

G	Get setting from reader; *data* contains a single character (0-255), which is the setting number. (The reader will respond with *d* and the setting value as a sequence of 8 ASCII hexadecimal digits or with *e*.) **Note:** See **Reader Settings** for possible reader settings.
I	Requests the reader to send its information string (no *data*). (The reader will respond with *i* or *e*.)
J	Requests the reader to restore settings to defaults (no *data*). (The reader will respond with *d* or *e*.)
J1	Complete restore of factory setup. Will overwrite the apps and settings.
L	Requests the reader to send a list of its stored files. *data* is: • (no *data*) or "0"; all non-hidden files. • "1"; hidden files (The reader will respond in the same manner as with the '(' command, each *z* packet containing a file name as a NUL-terminated string of printable ASCII characters.)
O	Set a bit (or bits) in a reader setting; *data* is as defined in the / command; the specified integer is ORed with the existing setting value. (The reader will respond with *d* or *e*.) **Note:** See **Reader Settings** for possible reader settings.
P	Put setting to reader; *data* is as defined in the / command; the specified integer replaces the existing setting value. (The reader will respond with *d* or *e*.) **Note:** See **Reader Settings** for possible reader settings.
Q	Clear a bit (or bits) in a reader setting; *data* is as defined in the / command; the ones-complement of the specified integer is AND'ed with the existing setting value. (The reader will respond with *d* or *e*.) **Note:** See **Reader Settings** for possible reader settings.
R	Requests that the previously sent packet be re-sent by the reader; *data* may specify a maximum packet size the receiver will accept: *data* is either empty or specifies a 16-bit big-endian unsigned integer (2 bytes). If *data* is empty or specifies a size less than 32 (the minimum packet size), the reader will use its preferred maximum packet size. Otherwise, it will use the specified max packet size (or less) and will fragment data across multiple smaller packets when necessary. (The reader will respond by re-sending its previous packet or with *e* if there was no previous packet. If the max data size has changed, it may resend the previous data in a sequence of more than one packet.)

T	Requests the current date and time (no *data*). (The reader will respond with *d* with *data* containing the date and time formatted as yyyy-mm-dd hh:mm:ss.) **Note:** On units without a battery-backed real-time clock, the date and time will reset to 2000-01-01 00:00:00 upon power-up.
U	Reserved for script engine.
W	Requests the reader to write its current settings from RAM to its non-volatile memory. (The reader will respond with *d* or *e*.)
Y	Acknowledge the receipt of a packet; *data* specifies the received packet number (one byte). (The reader will not respond.)
Z	Request the reader to reboot. *data* is: • *empty* or '0'; reboot the reader. • '1'; restart application. (The reader will respond with *d* or *e* before it reboots.)
^	Requests the reader to upload the specified stored file; *data* contains the file name, terminated with ASCII NUL. The reader will respond with: • 'g' packet containing *"filename<tab>(size)"* • 'z' packet(s) • 'd' packet containing "EOF<*tab*>(CRC16)" **Note:** *filename* "help" is reserved to send command information.
–	Causes the reader to wait for all buttons to be released and clear its event queue. (The reader will respond with *d* or *e*.)
|	Process *data* as a decoded string. (The reader will respond with *d* or *e*.)

Simple Protocol

The file is split into blocks of 236 or less bytes each and downloaded to the reader via *1, 2,* and *5* commands using the following sequence:

1. Send a *1* command to initialize the download.
2. Wait for a *d* or *e* response from the reader or a timeout.
   a. If timeout or *e* response, restart the sequence at step 1.
   b. If *d* response, continue to step 3.
3. Send a series of *2* commands, each with a portion of the file. (The reader will not send any response.)
4. Send a *5* command to end the download and install the file.
5. Wait for a *d, e,* or *f* response from the reader or a timeout.
   a. If *f* response or timeout, restart the sequence at step 1.
   b. If *e* response, repeat step 5.
   c. If *d* response, file download has completed successfully.

Note: The timeout will need to be increased from the normal response timeout to allow the firmware time to write the file to the flash memory.

Reader Settings

The host sets the reader settings using the */, O, P, Q,* and = commands and reads them using the *G*, **,**, and < commands.

For example, the following P command sets register 2C to the value C8.

P(2C)C8

Note: For two-digit setting numbers (i.e., settings 00 through fd), an alternative format may be used: in place of the parentheses and hexadecimal setting number, substitute a single character, which represents the setting number. The equivalent to the example above is P,C8 (the ASCII '**,**' character has the hexadecimal value 2c). (In certain circumstances, such as with text commands, "percent-encoding" may be used for encoding a character as a sequence consisting of the percent character followed by two hexadecimal digits. With percent-encoding, the example may be expressed as P%2CC8.)

In the **Reader Settings Table**, the **Reg** column is the setting number, in **hexadecimal**, to be used with the commands identified above. In the **Default** column, all values are in **hexadecimal** unless otherwise specified. To use decimal values in commands you must precede the data with a pound sign '**#**'. The following P command sets register 2C to the same value as the example above:

P(2C)#200

Since the single digit values of 0 through 9 are identical in decimal and hexadecimal, no indicator is needed.

Binary Dip Switch

Some registers are what Omron Microscan terms a ‘Binary Dip Switch’ where the value of each bit of the data string switches on or off some part of the behavior of that register. The bits are numbered from least significant to most (right to left). Each bit can be on or off (1 or 0).

An example of this is register 0B, ‘Codabar Checksum’. The following settings are possible:

Bit (R to L)	Controls	Value
0	Codabar Checksum Checking	0: Disabled
		1: Enabled
1	Strip Checksum from Output	0: Disabled
		1: Enabled

Given the settings above, the binary string turns Codabar Checksum ON and strips it from output.

Thus, the command to implement the settings above would be:

P(48)3

or

P(48)#03

Field of Interest

The reader optics are typically split into two separate fields - Field Of Interest 0 (FOI0) and Field Of Interest 1 (FOI1). In certain circumstances, these fields can be customized to the requirements of the user. In the default configuration of these fields FOI0 is the High Density (HD) field and FOI1 is the Wide (W) field.

At a given focus distance, the HD field is designed to read small, low-mil symbols while the Wide field is designed to pick up large, wide symbols.

This document will refer to FOI0 as HD and FOI1 as Wide.

Reader Settings Table

<table>
<tr><th>Reg</th><th>Setting Name</th><th>Default (Hex)</th><th>Comment</th></tr>
<tr><td>04</td><td>Continuous Illumination During Read</td><td>0</td><td>0: Minimal Illumination
1: Leave Illumination On Until End Read Cycle
Leave illumination on during read.</td></tr>
<tr><td>08</td><td>Reader Packet Format</td><td>1</td><td>1: Raw
2: Packet Mode Version 1
For example, USB “two-way” native:
1B: 5 (USB Native)
08: 2 (packet mode)
42: 1 (expect response)
Also see registers: 1B, 42</td></tr>
<tr><td>0A</td><td>NEC 2 of 5 Symbology</td><td>1</td><td>Binary Dip Switch
<table>
<tr><th>Bit</th><th>Controls</th><th>Value</th></tr>
<tr><td rowspan="2">0</td><td rowspan="2">NEC 2 of 5 Decoding</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">1</td><td rowspan="2">Checksum checking</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">2</td><td rowspan="2">Strip checksum from result</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">3</td><td rowspan="2">1 Digit Symbol Allowed</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">4</td><td rowspan="2">2 Digit Symbol Allowed</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
</table>
Note: All symbol lengths greater than 2 are always enabled when NEC 2 of 5 Decoding is enabled.</td></tr>
</table>

<table>
<tr><td>0B</td><td>Matrix 2 of 5 Symbology</td><td>1</td><td>Binary Dip Switch<table><tr><th>Bit</th><th>Controls</th><th>Value</th></tr><tr><td rowspan="2">0</td><td rowspan="2">Matrix 2 of 5 Decoding</td><td>0: Disabled</td></tr><tr><td>1: Enabled</td></tr><tr><td rowspan="2">1</td><td rowspan="2">Checksum checking</td><td>0: Disabled</td></tr><tr><td>1: Enabled</td></tr><tr><td rowspan="2">2</td><td rowspan="2">Strip checksum from result</td><td>0: Disabled</td></tr><tr><td>1: Enabled</td></tr><tr><td rowspan="2">3</td><td rowspan="2">1 Digit Symbol Allowed</td><td>0: Disabled</td></tr><tr><td>1: Enabled</td></tr><tr><td rowspan="2">4</td><td rowspan="2">2 Digit Symbol Allowed</td><td>0: Disabled</td></tr><tr><td>1: Enabled</td></tr></table>Note: All symbol lengths greater than 2 are always enabled when Matrix 2 of 5 Decoding is enabled.</td></tr>
<tr><td>0C</td><td>Telepen Symbology</td><td>1</td><td>0: Disabled
1: Enabled</td></tr>
<tr><td>0D</td><td>Enable Non-Square Data Matrix Symbology</td><td>0</td><td>0: Disabled
1: Enabled</td></tr>
<tr><td>0F</td><td>Targeting Control</td><td>1</td><td>0: Targeting Disabled
1: Targeting Enabled</td></tr>
<tr><td>16</td><td>Data Matrix Rectangular Symbology</td><td>0</td><td>0: Disabled
1: Enabled</td></tr>
<tr><td>19</td><td>Data Matrix Symbology</td><td>1</td><td>Binary Dip Switch<table><tr><th>Bit</th><th>Controls</th><th>Value</th></tr><tr><td rowspan="2">0</td><td rowspan="2">Data Matrix Decoding</td><td>0: Disabled</td></tr><tr><td>1: Enabled</td></tr><tr><td rowspan="2">1</td><td rowspan="2">Inverse Data Matrix Decoding</td><td>0: Disabled</td></tr><tr><td>1: Enabled</td></tr></table></td></tr>
</table>

1B	Communications Mode	8	1: RS232 serial 2: USB keyboard 5: USB Native (HID) 6: USB VComm 8: Dynamic (1 if decode is RS232; 2 if decode is USB) This setting is used in conjunction with settings 08 and 42 to configure the communication mode between standard “one-way” and “two-way” modes. For example, USB “two-way” native: 1b: 5 (USB Native) 08: 2 (packet mode) 42: 1 (expect response) **Note:** The following must be completed within 1 second. first output report with numlock set and capslock clear second output report with numlock set and capslock clear third output report with capslock set numlock clear fourth output report with capslock set numlock clear fifth output report with numlock set and capslock clear sixth output report with numlock set and capslock clear On the last output report comm protocol is set to raw mode, comm expect response is false and comm mode is USB Downloader mode. Also see registers: 08, 42
1C	Serial Baud Rate	1C200 (#115200)	All standard baud rates up #115200 • #9600 (2580) • #19200 (4B00) • #38400 (9600) • #57600 (E100) • #115200 (1C200)
1D	Serial Stop Bits	1	1: One 2: Two
1E	Serial Data Bits	8	7: Seven 8: Eight

<table>
<tr><td>22</td><td>Serial Parity</td><td>0</td><td>0: None
1: Odd
2: Even</td></tr>
<tr><td>26</td><td>Beep Volume (percent)</td><td>64 (#100)</td><td>Valid Range: 0 to 64 (#100) Percent
This is the current percentage of full volume potential.
Also see registers: 59, A7</td></tr>
<tr><td>29</td><td>PDF417 Symbology</td><td>1</td><td>0: Disabled
1: Enabled
Also see registers: 2A, CF</td></tr>
<tr><td>2A</td><td>MicroPDF417 Symbology</td><td>0</td><td>0: Disabled
1: Enabled
Also see registers: 29, CF</td></tr>
<tr><td>2B</td><td>QR Code Symbology</td><td>1</td><td>Binary Dip Switch<table><tr><th>Bit</th><th>Controls</th><th>Value</th></tr><tr><td rowspan="2">0</td><td rowspan="2">QR Code Decoding</td><td>0: Disabled</td></tr><tr><td>1: Enabled</td></tr><tr><td rowspan="2">1</td><td rowspan="2">Inverse QR Code Decoding</td><td>0: Disabled</td></tr><tr><td>1: Enabled</td></tr><tr><td rowspan="2">2</td><td rowspan="2">Micro QR Code Decoding</td><td>0: Disabled</td></tr><tr><td>1: Enabled</td></tr><tr><td rowspan="2">3</td><td rowspan="2">Inverse Micro QR Code Decoding</td><td>0: Disabled</td></tr><tr><td>1: Enabled</td></tr></table></td></tr>
</table>

<table>
<tr>
<td>2C</td>
<td>Idle Mode Countdown Timer (ms)</td>
<td>64 (#100)</td>
<td>Valid Range: 0 to 7FFFFFFF Milliseconds
Counts down to the change to Idle Mode.
The most significant bit (MSB) of the 32-bit register indicates whether this timer is enabled. Enable or Disable the timer by setting the MSB.
You can change the big directly by setting the register value (such as setting to #100) or you can change the value of just the MSB using the O (set), Q (clear) or / (toggle) bit commands. See Command Types for more information on these commands.
<table>
<tr><th>Action</th><th>Command</th></tr>
<tr><td>Enable</td><td>O(32)#-2147483648</td></tr>
<tr><td>Disable</td><td>Q(32)#-2147483648</td></tr>
<tr><td>Toggle</td><td>/(32)#-2147483648</td></tr>
</table>
Idle Mode is the time between the last user interaction with the imager (button press, etc.) or firmware interaction (communications, etc.) and Standby Mode. There are many user and firmware events that will reset the timer. Therefore, it may seem that the timer is longer than the value set.
This state: Active (highest power usage)
Next state: Idle</td>
</tr>
<tr>
<td>2D</td>
<td>Keyboard Maps</td>
<td>0</td>
<td>0: US English (without leading 0 in the ALT _ Number)
1: ASCII (ALT+number) - universal
2: Custom (requires user to download keyboard map)
3: US English (with leading 0 in the ALT + number for non-printable ASCII)
4: French Keyboard
5: German Keyboard
6: Japanese Keyboard
7: US English (with CTRL + char for non-printable ASCII)</td>
</tr>
</table>

34	Maximum Candidate Decodes Per Read	1	The Reader will process up to this number of codes per “read code” event. If there are more than this many codes in the field of view and within target tolerance, only the first ones will be decoded. For fastest performance with single codes, set to 1.
35	Button Stay-Down Time (ms)	0	Valid Range: 0 to 7FFFFFFF Milliseconds Keep processing the “read code” events for this amount of time (act as if the button stays down for this time)
36	Number of Control Frames Before Picture Capture	0	Valid Range: 0 to 7FFFFFFF Frames Number of frames captured and discarded before live picture to give the gain control time to adjust. Also see registers: 43, AC, AD, AE, AF
39	Trigger 1	3	The specified event is posted upon press of this button. The events are defined below: 0: No Action 1: Keep Awake 2: Show Target 3: Read In Both Fields (Default) 4: Default Event Selected By Hardware 5: Read In High Density field (FOI0). 6: Read In Wide field (FOI1). 7: Take Picture 8: Read In Most Recently Successful Field 255: Idle
40	Text Command Timeout (ms)	2AF8 (#11000)	Valid Range: 0 to 7FFFFFFF Milliseconds The maximum time during which a complete text command from Host must be received. (Pending text command data is discarded when the timeout is exceeded.)

<table>
<tr>
<td>41</td>
<td>Text Commands</td>
<td>8</td>
<td>
Binary Dip Switch
<table>
<tr><th>Bit</th><th>Controls</th><th>Value</th></tr>
<tr><td rowspan="2">0</td><td rowspan="2">Text Commands</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">1</td><td rowspan="2">Suppress Echo</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">2</td><td rowspan="2">Suppress Responses</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">3</td><td rowspan="2">Disable Text Commands but Enable Magic Sequence</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">4</td><td rowspan="2">Suppress URL Decode; See Below</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">5</td><td rowspan="2">Accept On Time-out</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
</table>
Magic Sequence:
The Magic Sequence is the string ";>PAx" where x is 1, 3, or 7 as defined above. This would normally be used in command text files, which would begin with the text-command-on sequence and end with the command to return to this special mode. For example:
;>PA7
;any desired commands here
PA8
Suppress URL Decode:
For example, if enabled, P%418 will not equal PA8. The % is not recognized as an escape character
Accept On Timeout:
</td>
</tr>
</table>

<table>
<tr><td>42</td><td>Expect Acknowledgement From Host</td><td>0</td><td>0: Reader doesn’t wait for acknowledge
1: Reader will retransmit data when Host doesn’t acknowledge receipt
This setting is used in conjunction with settings 1B and 42 to configure the communication mode between standard “one-way” and “two-way” modes.
For example, USB “two-way” native:
1B: 5 (USB Native)
08: 2 (packet mode)
42: 1 (expect response)
Also see registers: 08, 1B</td></tr>
<tr><td>43</td><td>JPEG Picture Quality (percent)</td><td>32 (#50)</td><td>Valid Range: 0 to 64 (#100) Percent
0: Raw Image (No JPEG Compression)
1 To 100: JPEG Compression Quality Percent
Also see registers: 36, AC, AD, AE, AF</td></tr>
<tr><td>45</td><td>Read Cycle Timeout</td><td>1F4 (#500)</td><td>Valid Range: 0 to FFFF (#65535) ms</td></tr>
<tr><td>47</td><td>Maxicode Symbology</td><td>0</td><td>Binary Dip Switch
<table>
<tr><th>Bit</th><th>Controls</th><th>Value</th></tr>
<tr><td rowspan="2">0</td><td rowspan="2">Maxicode Decoding, Mode 0</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">1</td><td rowspan="2">Maxicode Decoding, Mode 1</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">2</td><td rowspan="2">Maxicode Decoding, Mode 2</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">3</td><td rowspan="2">Maxicode Decoding, Mode 3</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">4</td><td rowspan="2">Maxicode Decoding, Mode 4</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">5</td><td rowspan="2">Maxicode Decoding, Mode 5</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">6</td><td rowspan="2">Maxicode Decoding, Mode 6</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
</table></td></tr>
</table>

<table>
<tr><td>48</td><td>Codabar Checksum</td><td>0</td><td>Binary Dip Switch
<table><tr><th>Bit</th><th>Controls</th><th>Value</th></tr><tr><td rowspan="2">0</td><td rowspan="2">Codabar Checksum Checking</td><td>0: Disabled</td></tr><tr><td>1: Enabled</td></tr><tr><td rowspan="2">1</td><td rowspan="2">Strip Checksum from Output</td><td>0: Disabled</td></tr><tr><td>1: Enabled</td></tr></table></td></tr>
<tr><td>49</td><td>Code 39 Symbology</td><td>0</td><td>0: Disabled
1: Enabled
Code 39 Full ASCII Decoding</td></tr>
<tr><td>4A</td><td>Composite Codes</td><td>0</td><td>0: Disabled
1: Enabled
Composite Code Decoding
Also see register: D8</td></tr>
<tr><td>4B</td><td>Postal Code Symbology</td><td>0</td><td>0: Disabled
1: Enabled</td></tr>
<tr><td>4C</td><td>GS1 Symbology</td><td>1F (#31)</td><td>Binary Dip Switch
<table><tr><th>Bit</th><th>Controls</th><th>Value</th></tr><tr><td rowspan="2">0</td><td rowspan="2">GS1 Expanded decoding</td><td>0: Disabled</td></tr><tr><td>1: Enabled</td></tr><tr><td rowspan="2">1</td><td rowspan="2">GS1 Expanded Stacked decoding</td><td>0: Disabled</td></tr><tr><td>1: Enabled</td></tr><tr><td rowspan="2">2</td><td rowspan="2">GS1 Limited decod-ing</td><td>0: Disabled</td></tr><tr><td>1: Enabled</td></tr><tr><td rowspan="2">3</td><td rowspan="2">GS1-14 and GS1-14 Truncated decoding</td><td>0: Disabled</td></tr><tr><td>1: Enabled</td></tr><tr><td rowspan="2">4</td><td rowspan="2">GS1-14 Stacked and GS1-14 Stacked Omnidirectional decoding</td><td>0: Disabled</td></tr><tr><td>1: Enabled</td></tr></table></td></tr>
<tr><td>4D</td><td>UPC Expansion</td><td>0</td><td>0: Disabled
1: Enabled
Also see registers: 4E, 6A, 74</td></tr>
</table>

<table>
<tr><td>4E</td><td>UPC Supplemental</td><td>0</td><td>0: Disabled
1: Enabled
Also see registers: 4D, 6A, 74</td></tr>
<tr><td>4F</td><td>MSI Plessey Symbology</td><td>1</td><td>Binary Dip Switch<table>
<tr><th>Bit</th><th>Controls</th><th>Value</th></tr>
<tr><td rowspan="2">0</td><td rowspan="2">MSI Plessey</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="8">[3:1]</td><td rowspan="8">Checksum</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td>1: 1 mod 10</td></tr>
<tr><td>2: mod 10 and mod 11</td></tr>
<tr><td>3: 2 mod 10</td></tr>
<tr><td>5: 1 mod 10 strip cs</td></tr>
<tr><td>6: mod 10 and mod 11 strip cs</td></tr>
<tr><td>7: 2 mod 10 strip cs</td></tr>
<tr><td rowspan="2">4</td><td rowspan="2">Improved Bounds</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
</table></td></tr>
<tr><td>50</td><td>Aztec Symbology</td><td>1</td><td>Binary Dip Switch<table>
<tr><th>Bit</th><th>Controls</th><th>Value</th></tr>
<tr><td rowspan="2">0</td><td rowspan="2">Aztec decoding</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">1</td><td rowspan="2">Inverse Aztec decoding</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
</table></td></tr>
<tr><td>53</td><td>Decoder HD field (FOI0) Width</td><td>280 (#640)</td><td>Valid Range: 1 to 640 pixels
Decoder uses only the specified pixel width in the HD field (FOI0).
Also see registers: 54, 98, 99</td></tr>
<tr><td>54</td><td>Decoder HD field (FOI0) Height</td><td>3C0 (#960)</td><td>Valid Range: 1 to 960 pixels
Decoder uses only the specified pixel height in the HD field (FOI0).
Also see registers: 53, 98, 99</td></tr>
</table>

<table>
<tr><td>55</td><td>Notify Of Read Failure</td><td>0</td><td>0: Disabled
1: Send “r” packet on no-read (See “r” packet in Packet Data.)
0x100xx: post event on no-read, where the lower 8 bits specify the event number. For example, 0x10009 to post Event 0x09.</td></tr>
<tr><td>59</td><td>Beep Duration</td><td>64 (#100)</td><td>Valid Range: 0 to 7FFFFFFF Milliseconds
Also see registers: 26, A7</td></tr>
<tr><td>66</td><td>Bypass Illumination</td><td>0</td><td>Valid Range: 0 to 64 (#100) percent</td></tr>
<tr><td>6A</td><td>UPC Symbology</td><td>1</td><td>0: Disabled
1: Enabled
Also see registers: 4D, 4E, 74</td></tr>
<tr><td>6B</td><td>Code 39 Symbology</td><td>1</td><td>0: Disabled
1: Enabled
Also see register: 70</td></tr>
<tr><td>6C</td><td>Code 93 Symbology</td><td>1</td><td>0: Disabled
1: Enabled</td></tr>
<tr><td>6D</td><td>Code 128 Symbology</td><td>1</td><td>0: Disabled
1: Enabled</td></tr>
<tr><td>6E</td><td>Interleaved 2 Of 5 Symbology</td><td>1</td><td>0: Disabled
1: Enabled
Also see registers: 71, C9</td></tr>
<tr><td>6F</td><td>Codabar Symbology</td><td>1</td><td>0: Disabled
1: Enabled</td></tr>
<tr><td>70</td><td>Code 39 Checksum</td><td>0</td><td>Binary Dip Switch<table><tr><th>Bit</th><th>Controls</th><th>Value</th></tr><tr><td rowspan="2">0</td><td rowspan="2">Code 39 Checksum Checking</td><td>0: Disabled</td></tr><tr><td>1: Enabled</td></tr><tr><td rowspan="2">1</td><td rowspan="2">Strip Checksum from Output</td><td>0: Disabled</td></tr><tr><td>1: Enabled</td></tr></table>Also see register: 6B</td></tr>
<tr><td>71</td><td>Interleaved 2 Of 5 Checksum</td><td>0</td><td>Binary Dip Switch<table><tr><th>Bit</th><th>Controls</th><th>Value</th></tr><tr><td rowspan="2">0</td><td rowspan="2">Interleaved 2 of 5 Checksum Checking</td><td>0: Disabled</td></tr><tr><td>1: Enabled</td></tr><tr><td rowspan="2">1</td><td rowspan="2">Strip Checksum from Output</td><td>0: Disabled</td></tr><tr><td>1: Enabled</td></tr></table>Also see register: 6E, C9</td></tr>
</table>

<table>
<tr><td>74</td><td>UPC Short Margin</td><td>1</td><td>0: Disabled
1: Enabled
Also see registers: 4D, 4E, 6A</td></tr>
<tr><td>78</td><td>Settings Lock</td><td>1</td><td>1: Settings unlocked
3: Settings locked (except settings Lock)</td></tr>
<tr><td>85</td><td>Trioptic Options</td><td>0</td><td>Binary Dip Switch
<table><tr><th>Bit</th><th>Controls</th><th>Value</th></tr><tr><td rowspan="2">0</td><td rowspan="2">Trioptic Decoding, Normal Quiet Zones</td><td>0: Disabled</td></tr><tr><td>1: Enabled</td></tr><tr><td rowspan="2">1</td><td rowspan="2">Allow Short Quiet Zones</td><td>0: Disabled</td></tr><tr><td>1: Enabled</td></tr><tr><td rowspan="2">2</td><td rowspan="2">No Quiet Zones (requires firmware version 3280+)</td><td>0: Disabled</td></tr><tr><td>1: Enabled</td></tr></table></td></tr>
<tr><td>86</td><td>Motion Detection: Event</td><td>3</td><td>Valid Range: (see register 39)
Motion detection is enabled by setting register C4 to 0xF0.
This register is reset to 0 (disabled) when register C4 is changed away from 0xF0.
When motion is detected, this event is posted.
See register 39 for list of events.
Also see registers: 20E</td></tr>
<tr><td>87</td><td>Motion Sensitivity</td><td>5</td><td>Valid Range: 0 to FFFF (#65535) ms
Also see registers: 86, 20E</td></tr>
<tr><td>93</td><td>Suppress Beep On Decode</td><td>0</td><td>0: Beep indicating decode before JavaScript processing
1: Call JavaScript without beeping to indicate decode
Normally, the Reader beeps as soon as decodes are read and processes them via JavaScript if necessary after the beep. To enable JavaScript to control the beep feedback, change this setting to 1; this will suppress the beep; the JavaScript would typically beep if the decode is valid or start another read cycle if it isn’t.
This setting does not suppress beeps for anything but a successful decode event.</td></tr>
</table>

98	Decoder HD field (FOI0) X Offset	0	Valid Range: 0 to 639 pixels Decoder uses the pixels after the specified pixel offset in the HD field (FOI0). Also see registers: 53, 54, 99
99	Decoder HD field (FOI0) Y Offset	0	Valid Range: 0 to 959 pixels Decoder uses the pixels after the specified pixel offset in the HD field (FOI0). Also see registers: 53, 54, 98
9A	Decoder Wide field (FOI1) X Offset	0	Valid Range: 0 to 639 pixels Decoder uses the pixels after the specified pixel offset in the Wide field (FOI1). Also see registers: 9B, C7, C8
9B	Decoder Wide field (FOI1) Y Offset	0	Valid Range: 0 to 959 pixels Decoder uses the pixels after the specified pixel offset in the Wide field (FOI1). Also see registers: 9A, C7, C8
9D	Target Tolerance (percent)	640 (#1600)	Valid Range: 0 to 7FFFFFFF Percent For the Reader to accept a code, the target dot must be within the code rectangle or in proximity to the symbol. The nearness is defined as this percentage of the code's smaller dimension. For example, with a 10 x 20 mm code and a setting of 150 (%), the target dot must be within 15 mm of the code. Any value over 1000 is considered infinite tolerance, and no target checking is performed.
A2	Default Event Delay (ms)	64 (#100)	Valid Range: 0 to 7FFFFFFF Milliseconds The Reader will pause for this amount of time between each posting of the default event (used with "continuous read" mode). Also see register C4
A7	Beep Pulse Separation (ms)	64 (#100)	Valid Range: 0 to 7FFFFFFF Milliseconds The spacing in milliseconds between beeps. Also see registers: 26, 59
A8	Bypass Exposure	19 (#25)	Valid Range: 0 to 64 (#100) percent Also see register: D1
A9	Bypass Gain	5F (#95)	Valid Range: 0 to 64 (#100) percent Also see registers: D1

AC	Wide field (FOI1) Picture Window Left Position	0	Specify left edge of window used with “take picture.” The position and size are relative to the virtual image (i.e., not the rotated physical image). **Note:** Overall image is 960 pixels by 1280 pixels. Upper half is Wide field (FIO1); lower half is High Density (FOI0). Also see registers: 36, 43, AD, AE, AF
AD	Wide field (FOI1) Picture Window Upper Position	0	Specify upper edge of window used with “take picture.” The position and size are relative to the virtual image (i.e., not the rotated physical image). **Note:** Overall image is 960 pixels by 1280 pixels. Upper half is Wide field (FIO1); lower half is High Density (FOI0). Also see registers: 36, 43, AC, AE, AF
AE	Wide field (FOI1) Picture Window Width	500 (#1280)	Specify width of window used with “take picture.” The position and size are relative to the virtual image (i.e., not the rotated physical image). **Note:** Overall image is 960 pixels by 1280 pixels. Upper half is Wide field (FIO1); lower half is High Density (FOI0). Also see registers: 36, 43, AC, AD, AF
AF	Wide field (FOI1) Picture Window Height	3C0 (#960)	Specify height of window used with “take picture.” The position and size are relative to the virtual image (i.e., not the rotated physical image). **Note:** Overall image is 960 pixels by 1280 pixels. Upper half is Wide field (FIO1); lower half is High Density (FOI0). Also see registers: 36, 43, AC, AD, AE
B0	Target On Before Picture (ms)	3E8 (#1000)	Valid Range: 0 to 7FFFFFFF Milliseconds 0: Target off before picture capture Also see registers: 36, 43, AC, AD, AE
B3	Number Of Retries Before Reader Gives Up Sending Packet	3	**Note:** The value 1 is defined as the original send attempt but no resends. Also see register: 42
BF	USB Keyboard Poll Rate	A (#10)	Valid Range: 1 to FF (#255) Milliseconds The Host is requested to poll the USB device at the specified period.

C4	Default (Continuous) Event	FF (#255)	The default value of FF (idle event) disables “continuous scanning”. Use one of the read events to enable “continuous scanning.” See setting 39 for the list of events. When no button is pressed but the Reader is still in active mode (i.e., not power-saving idle or sleep modes), this event will be posted. Also see register: 39
C7	Decoder Wide field (FOI1) Width	280 (#640)	Valid Range: 1 to 280 (#640) pixels Decoder uses only the specified pixel height in the Wide field (FOI1). Also see registers: 9A, 9B, C8
C8	Decoder Wide field (FOI1) Height	3C0 (#960)	Valid Range: 1 to 3C0 (#960) pixels Decoder uses only the specified pixel height in the Wide field (FOI1). Also see registers: 9A, 9B, C7
C9	Interleaved 2 Of 5 Lengths	0	FFFFFFFC: 2 and 4 digit disabled FFFFFFFD: 2 digit enabled FFFFFFFE: 4 digit enabled Also see registers: 6E, 71
CF	Macro PDF417 Symbology	0	0: Disabled 1: Enabled Also see registers: 29, 2A
D1	AGC Mode	0	0: Dynamic 2: Bypass
D8	Composite Codes Require Both Elements	1	0: Accept any composite element 1: Only accept composite codes if both elements could be decoded. Also see register 4A
EB	Maximum Reader To Host Packet Data Size	4000 (#16384)	Valid Range: 1 to 4000 (#16384)
EC	Host Acknowledgement Time Limit Multiplier (ms)	F (#15)	Valid Range: 0 to 7FFFFFFF Milliseconds When Expect Acknowledgement From Host (register 42) is nonzero, the Reader will wait up to Host Acknowledgement Time Limit (register 37) + dataSize * Host Acknowledgement Time Limit Multiplier (register EC) milliseconds to receive an acknowledgement from the Host.

ED	Prefix Decode Result With AIM Symbology Identifiers	0	0: Don't prefix with AIM identifier 1: Prefix decode result with ISO/IEC standard 15424/AIM symbology identifier
F0	Allow Code 128 Short Margin	1	0: Disabled 1: Enabled
F6	Code 39 Short Margin	1	0: Disallow short margin Code 39 symbol decoding 1: Allow short margin Code 39 symbol decoding
F8	PharmaCode Symbology	0	0: Disabled 1: Enabled
F9	PharmaCode Bar Count	1004 (#4100)	Valid Range: Each 8 bits can be 04 to 10 (#16) Bit 0 – Bit 7: min bar count, 04 to 10 (#16) Bit 9 – Bit 15: max bar count, 04 to 10 (#16)
FA	PharmaCode Min Value	F (#15)	Valid Range: F (#15) to 1FFFE (#131070)
FB	PharmaCode Max Value	1FFFE (#131070)	Valid Range: F (#15) to 1FFFE (#131070)
10B	Enable JavaScript	1	0: Disabled 1: Enabled When set to 0 installed scripts are disabled. This can be useful from boot mode for recovering the unit if a non-responsive script is installed.
10D	Data Matrix Symbol Identification Effort	2	0: Normal effort 1: Increase effort 2: Max effort Increases the decoder's effort to find a Data Matrix symbol in an image.
12C	Data Matrix Improvement	1	0: Disabled 1: Enabled Improves the decoding capability of the Reader on low contrast or pixelated Data Matrix bar codes

<table>
<tr><td>12D</td><td>Hong Kong 2 Of 5 Symbology</td><td>0</td><td>Binary Dip Switch<table><tr><th>Bit</th><th>Controls</th><th>Value</th></tr><tr><td rowspan="2">0</td><td rowspan="2">Hong Kong 2 of 5 decoding</td><td>0: Disabled</td></tr><tr><td>1: Enabled</td></tr><tr><td rowspan="2">1</td><td rowspan="2">1 Digit Symbol Allowed</td><td>0: Disabled</td></tr><tr><td>1: Enabled</td></tr><tr><td rowspan="2">2</td><td rowspan="2">2 Digit Symbol Allowed</td><td>0: Disabled</td></tr><tr><td>1: Enabled</td></tr></table></td></tr>
<tr><td>137</td><td>PDF417 Handle Invalid Shift</td><td>0</td><td>0: Disabled
1: Enabled
Allows the decoding of PDF417 bar codes that were improperly encoded</td></tr>
<tr><td>159</td><td>Ignore Duplicate Code (ms)</td><td>0</td><td>Valid Range: 0 to 7FFFFFF Milliseconds
Consecutive duplicate codes (i.e., codes that contain the same data) are blocked for this amount of time (in milliseconds). 0 turns off blocking of duplicate codes.</td></tr>
<tr><td>1D7</td><td>Morphology</td><td>0</td><td>Binary Dip Switch<table><tr><th>Bit</th><th>Controls</th><th>Value</th></tr><tr><td rowspan="3">1-0</td><td rowspan="3">Technique</td><td>0: None</td></tr><tr><td>1: Erode</td></tr><tr><td>2: Dilate</td></tr><tr><td rowspan="3">3-2</td><td rowspan="3">Size</td><td>0: Small 3x3</td></tr><tr><td>1: Med. 5x5</td></tr><tr><td>2: Large 7x7</td></tr></table></td></tr>
<tr><td>1D8</td><td>BC412 Status</td><td>1</td><td>0: Disabled
1: Enabled</td></tr>
<tr><td>1D9</td><td>UPC/EAN Status</td><td>1</td><td>0: Disabled
1: Enabled</td></tr>
</table>

<table>
<tr><td>1DC</td><td>Pharmacode Settings</td><td>28A40</td><td>Binary Dip Switch<table>
<tr><th>Bit</th><th>Controls</th><th>Value</th></tr>
<tr><td rowspan="2">0</td><td rowspan="2">Direction</td><td>0: Forward</td></tr>
<tr><td>1: Reverse</td></tr>
<tr><td rowspan="2">1</td><td rowspan="2">Fixed Symbol Length Status</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="4">3-2</td><td rowspan="4">Bar Width Status</td><td>0 = Mixed</td></tr>
<tr><td>1 = All Narrow</td></tr>
<tr><td>2 = All Wide</td></tr>
<tr><td>3 = Use Fixed Threshold</td></tr>
<tr><td>8-4</td><td>Minimum Number of Bars Value</td><td>4-10 (#16) (Default 4)</td></tr>
<tr><td>13-9</td><td>Fixed Symbol Length Value</td><td>1-10 (16) (Default 5)</td></tr>
<tr><td>29-14</td><td>Fixed Bar Width Threshold</td><td>0-FFFF (#65535) (Default 10)</td></tr>
</table></td></tr>
<tr><td>1EA</td><td>Good Read Duration (ms)</td><td>C8 (#200)</td><td>Valid Range: 0 to 7FFFFFF Milliseconds</td></tr>
<tr><td>1EB</td><td>Decoder Data Matrix Module Size</td><td>32 (#50)</td><td></td></tr>
<tr><td>20E</td><td>Motion Detection: Start Delay (ms)</td><td>0</td><td>Valid Range: 0 to 7FFFFFFF Milliseconds
A built-in delay of 200 ms prevents motion detect to detect motion right after a successful decode. This allows the bar code to be removed without triggering a new decode. Use this to add an additional delay amount.
Also see register: 86</td></tr>
</table>

CRB System

The CRB system is a convenient method for creating and maintaining a set of commands that can be easily sent to the reader. These CRB files can be created in any text editor with the file extension of .crb. The CRB system accepts all of the valid *text commands*. The most commonly used commands are *J*, *N*, *P*, and ~. There should be one command per line. The CRB file may contain empty lines and comments as well.

The .crb files can be sent directly to the reader using the normal file transfer. As CRB files are just a list of *text commands*, they can also be sent by a serial terminal program. **Note: if using a serial terminal program the reader will first need to be set to "text command mode"; see Text Commands.**

You can request a copy of all MS-2D configuration settings in .crb format.

Example CRC16 C Code

CRC16.h:

```
// crc16.h
#ifndef crc16_h
#define crc16_h
#include <stdint.h>
#include <stddef.h>
#ifdef __cplusplus
extern "C" {
#endif
typedef uint16_t crc_t;
crc_t crc
(       crc_t initialCrc
        , const unsigned char* bufPtr
        , size_t length
);
#ifdef __cplusplus
} // extern "C"
#endif
#endif
/*eof*/
```

CRC16.c:

```
// crc16.c
#include <crc16.h>
crc_t crc
(         crc_t initialCrc
          , const unsigned char* p
          , size_t n
)
{
          enum
          {
                crcBits = 16,
                charBits = 8,
                diffBits = crcBits - charBits
          };
          crc_t c = initialCrc;
          #include "crc16tab.h"
          while( n-- )
                c = (c << charBits) ^ crcTab[( c >> diffBits ) ^ *p++];
          return c;
}
/*eof*/
```

CRC16tab.h:

```
/* crc16tab.h
 * crc16 table of partial remainders generated by
 * mkcrctab.c with polynomial 1021.
 * included only from within crc() function in file crc16.c
 */
static const crc_t crcTab[] =
{
    0x0000, 0x1021, 0x2042, 0x3063, 0x4084, 0x50a5, 0x60c6, 0x70e7,
    0x8108, 0x9129, 0xa14a, 0xb16b, 0xc18c, 0xd1ad, 0xe1ce, 0xf1ef,
    0x1231, 0x0210, 0x3273, 0x2252, 0x52b5, 0x4294, 0x72f7, 0x62d6,
    0x9339, 0x8318, 0xb37b, 0xa35a, 0xd3bd, 0xc39c, 0xf3ff, 0xe3de,
    0x2462, 0x3443, 0x0420, 0x1401, 0x64e6, 0x74c7, 0x44a4, 0x5485,
    0xa56a, 0xb54b, 0x8528, 0x9509, 0xe5ee, 0xf5cf, 0xc5ac, 0xd58d,
    0x3653, 0x2672, 0x1611, 0x0630, 0x76d7, 0x66f6, 0x5695, 0x46b4,
    0xb75b, 0xa77a, 0x9719, 0x8738, 0xf7df, 0xe7fe, 0xd79d, 0xc7bc,
    0x48c4, 0x58e5, 0x6886, 0x78a7, 0x0840, 0x1861, 0x2802, 0x3823,
    0xc9cc, 0xd9ed, 0xe98e, 0xf9af, 0x8948, 0x9969, 0xa90a, 0xb92b,
    0x5af5, 0x4ad4, 0x7ab7, 0x6a96, 0x1a71, 0x0a50, 0x3a33, 0x2a12,
    0xdbfd, 0xcbdc, 0xfbbf, 0xeb9e, 0x9b79, 0x8b58, 0xbb3b, 0xab1a,
    0x6ca6, 0x7c87, 0x4ce4, 0x5cc5, 0x2c22, 0x3c03, 0x0c60, 0x1c41,
    0xedae, 0xfd8f, 0xcdec, 0xddcd, 0xad2a, 0xbd0b, 0x8d68, 0x9d49,
    0x7e97, 0x6eb6, 0x5ed5, 0x4ef4, 0x3e13, 0x2e32, 0x1e51, 0x0e70,
    0xff9f, 0xefbe, 0xdfdd, 0xcffc, 0xbf1b, 0xaf3a, 0x9f59, 0x8f78,
    0x9188, 0x81a9, 0xb1ca, 0xa1eb, 0xd10c, 0xc12d, 0xf14e, 0xe16f,
    0x1080, 0x00a1, 0x30c2, 0x20e3, 0x5004, 0x4025, 0x7046, 0x6067,
    0x83b9, 0x9398, 0xa3fb, 0xb3da, 0xc33d, 0xd31c, 0xe37f, 0xf35e,
    0x02b1, 0x1290, 0x22f3, 0x32d2, 0x4235, 0x5214, 0x6277, 0x7256,
    0xb5ea, 0xa5cb, 0x95a8, 0x8589, 0xf56e, 0xe54f, 0xd52c, 0xc50d,
    0x34e2, 0x24c3, 0x14a0, 0x0481, 0x7466, 0x6447, 0x5424, 0x4405,
    0xa7db, 0xb7fa, 0x8799, 0x97b8, 0xe75f, 0xf77e, 0xc71d, 0xd73c,
    0x26d3, 0x36f2, 0x0691, 0x16b0, 0x6657, 0x7676, 0x4615, 0x5634,
    0xd94c, 0xc96d, 0xf90e, 0xe92f, 0x99c8, 0x89e9, 0xb98a, 0xa9ab,
    0x5844, 0x4865, 0x7806, 0x6827, 0x18c0, 0x08e1, 0x3882, 0x28a3,
    0xcb7d, 0xdb5c, 0xeb3f, 0xfb1e, 0x8bf9, 0x9bd8, 0xabbb, 0xbb9a,
    0x4a75, 0x5a54, 0x6a37, 0x7a16, 0x0af1, 0x1ad0, 0x2ab3, 0x3a92,
    0xfd2e, 0xed0f, 0xdd6c, 0xcd4d, 0xbdaa, 0xad8b, 0x9de8, 0x8dc9,
    0x7c26, 0x6c07, 0x5c64, 0x4c45, 0x3ca2, 0x2c83, 0x1ce0, 0x0cc1,
    0xef1f, 0xff3e, 0xcf5d, 0xdf7c, 0xaf9b, 0xbfba, 0x8fd9, 0x9ff8,
    0x6e17, 0x7e36, 0x4e55, 0x5e74, 0x2e93, 0x3eb2, 0x0ed1, 0x1ef0,
};

/*eof*/
```

Appendix E — Communications Protocol

Communications Protocol Command Table

Protocol Command (Mnemonic displayed on menu)	Control Characters (Entered in menu or serial command)	Hex Value	Effect of Command
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 F — ASCII Table

Dec	Hex	Mne	Ctrl
00	00	NUL	^@
01	01	SOH	^A
02	02	STX	^B
03	03	ETX	^C
04	04	EOT	^D
05	05	ENQ	^E
06	06	ACK	^F
07	07	BEL	^G
08	08	BS	^H
09	09	HT	^I
10	0A	LF	^J
11	0B	VT	^K
12	0C	FF	^L
13	0D	CR	^M
14	0E	SO	^N
15	0F	SI	^O
16	10	DLE	^P
17	11	DC1	^Q
18	12	DC2	^R
19	13	DC3	^S
20	14	DC4	^T
21	15	NAK	^U
22	16	SYN	^V
23	17	ETB	^W
24	18	CAN	^X
25	19	EM	^Y
26	1A	SUB	^Z
27	1B	ESC	^[
28	1C	FS	^\
29	1D	GS	^]
30	1E	RS	^^
31	1F	US	^_

Dec	Hex	Ch
32	20	SP
33	21	!
34	22	“
35	23	#
36	24	$
37	25	%
38	26	&
39	27	'
40	28	(
41	29	)
42	2A	*
43	2B	+
44	2C	,
45	2D	-
46	2E	.
47	2F	/
48	30	0
49	31	1
50	32	2
51	33	3
52	34	4
53	35	5
54	36	6
55	37	7
56	38	8
57	39	9
58	3A	:
59	3B	;
60	3C	<
61	3D	=
62	3E	>
63	3F	?

Dec	Hex	Ch
64	40	@
65	41	A
66	42	B
67	43	C
68	44	D
69	45	E
70	46	F
71	47	G
72	48	H
73	49	I
74	4A	J
75	4B	K
76	4C	L
77	4D	M
78	4E	N
79	4F	O
80	50	P
81	51	Q
82	52	R
83	53	S
84	54	T
85	55	U
86	56	V
87	57	W
88	58	X
89	59	Y
90	5A	Z
91	5B	[
92	5C	\
93	5D	]
94	5E	^
95	5F	_

Dec	Hex	Ch
96	60	`
97	61	a
98	62	b
99	63	c
100	64	d
101	65	e
102	66	f
103	67	g
104	68	h
105	69	i
106	6A	j
107	6B	k
108	6C	l
109	6D	m
110	6E	n
111	6F	o
112	70	p
113	71	q
114	72	r
115	73	s
116	74	t
117	75	u
118	76	v
119	77	w
120	78	x
121	79	y
122	7A	z
123	7B	{
124	7C	|
125	7D	}
126	7E	~
127	7F	D

Appendix G — Maintenance

The MS-2D provides reliable and efficient operation with a minimum of care. Although specific maintenance is not required, the following periodic checks ensure dependable operation.

Cleaning the MS-2D Window

The MS-2D has a clear, anti-reflective coated optical window that protects the illumination system and optics. The window should be clean to allow optimum performance. The MS-2D uses technology that is much like a digital camera, and marks or debris on the window will interfere with image captures. Avoid touching the coated surface, as fingerprints may impede decode performance.

In many cases the window can be cleaned by wiping with a lint-free lens cloth to remove dust, debris, and fingerprints. Care should be taken not to apply too much pressure, as a trapped particle may scratch the window.

If cleaning of the window becomes necessary, follow this procedure:

- Use a minimal amount of Isopropyl Alcohol to dampen. Do not saturate the surface, as this may cause streaking.
- Drag the moistened cotton swab, cotton ball, or soft, clean cloth across the coated surface. Do not rub.
- Repeat this procedure until no contaminants remain.

  Note: Many products designed for cleaning plastic lens eyewear, such as pre-moistened towelettes or lens cloths, can be used to clean the MS-2D window.

Cleaning the MS-2D Housing

If the housing becomes dirty, clean it with a soft, non-abrasive cloth that has been moistened with water. A mild detergent may be used to clean the housing, but the detergent should then be rinsed away with a water-moistened cloth.

Caution: Do not submerge the MS-2D in water. The housing is not watertight. Do not use abrasive cloths or tissues on the MS-2D window — abrasive cloths or tissues may scratch the window.

Appendix H — Optimizing the MS-2D for Low-Power Applications

Achieving low power consumption with the MS-2D requires that certain setup and configuration values be programmed into the engine, as well as certain protocols that need to be followed to wake up from sleep mode. This appendix describes these operations.

Configuration

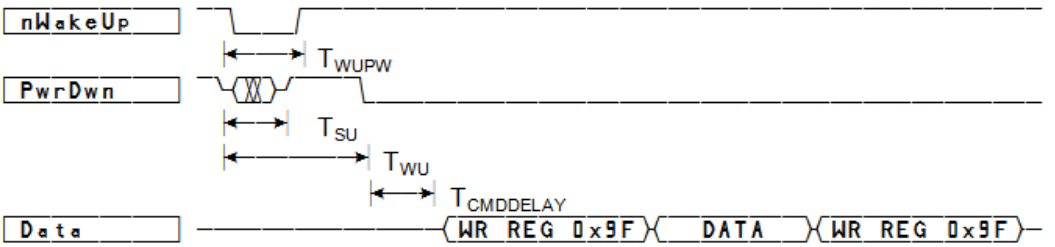
To configure the MS-2D for low power operation, various registers can be set to adjust sleep and wake timing. Register 0x9F controls the duration between a completed read operation and when the unit goes to sleep. It should be written with a non-zero value that is as small as possible given the application to minimize the idle time between scan completion and the MS-2D entering the sleep mode.

Communications from Sleep Mode

The host needs to assert the nWakeUp pin and wait for the PwrDwn pin to be negated before the host can communicate with the MS-2D. In conjunction with programming register 0x9F and this operation, the time that the MS-2D remains awake can be minimized.

Long-Term Communication

The following timing diagram shows the general process for waking up and communicating with the MS-2D on a long-term basis. This includes configuration and performing firmware upgrades on the engine.

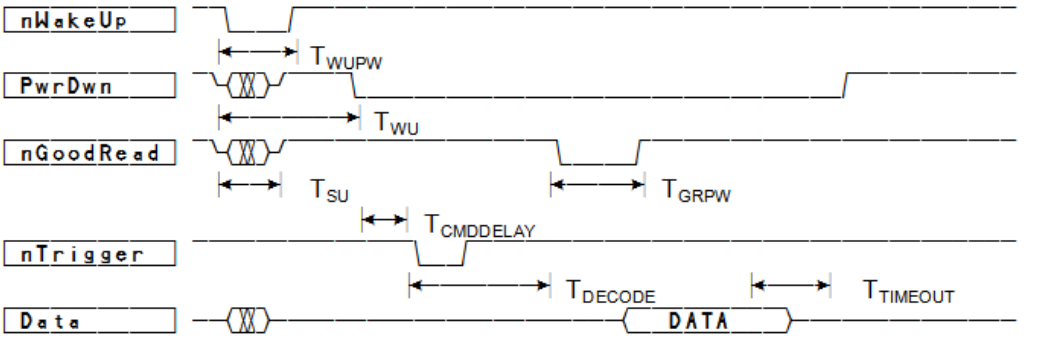


Wake-Up Communication Timing Diagram

1. Host asserts nWakeUp for at least TWUPW, and is optionally held low until communication with the MS-2D is finished.
2. After TWU, MS-2D PwrDwn negates.
3. After TCMDDELAY, host writes register 0x9F with 0x80000000 to disable sleep timer.
4. Host communicates with MS-2D.
5. Host writes register 0x9F with appropriate value to enable sleep timer.

Rapid Scanning and Returning to Sleep

The following diagram shows the timing diagram for scanning a code when the unit is in sleep mode. When register 0x9F is minimized, the unit will consume the minimum amount of power possible.



Wake-Up Code read Timing Diagram

1. Host asserts nWakeUp.
2. Output signals are invalid for TSU.
3. After TWU, MS-2D negates PwrDwn.
4. After TCMDDELAY, Host can assert nTrigger.
5. MS-2D decodes symbol and asserts nGoodRead.
6. MS-2D transmits data to host.
7. After MS-2D transmits data, engine goes to sleep after TTIMEOUT period programmed in register 0x9F.

Timing Specifications

Signal	Description	Min	Typ	Max	Units
TSU	Time between nWakeUp asserted and outputs valid	5	5	10	msec
TWU	Time between nWakeUp asserted and MS-2D ready			120	msec
TWUPW	nWakeUp pulse width	10	20		msec
TCMDDELAY	Time between MS-2D ready and when nTrigger can be asserted		0	1	msec
TDECODE	Time between nTrigger asserted and nGoodRead asserted (decode time)		100		msec
TGRPW	nGoodRead pulse width	See Note 1			
TTIMEOUT	Time between data transfer and sleep state	See Note 2			msec

Note 1: TGRPW is programmable using command register 0x??
Note 2: TTIMEOUT is programmable using command register 0x9F

Appendix I — 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.

ADC — See **Analog-to-Digital Converter**.

A/D Converter — See **Analog-to-Digital Converter**.

AGC — See **Automatic Gain Control**.

Ambient Light — Light which is present in the environment of the front end of a reader and generated from outside sources. This light, unless used for actual illumination, will be treated as background noise by the reader.

Analog — A smooth, continuous voltage or current signal or function whose magnitude (value) is the information.

Analog-to-Digital Converter (A/D Converter or **ADC)** — 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.

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.

CCD — See **Charge-Coupled Device**.

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.

Connector — A plug or socket on a device or cable providing in/out connectivity for various circuits and pins.

Concentrator — Intermediary device that relays data from readers to a host and commands from the host to the readers or other devices.

DAC — See **Digital-to-Analog Converter**.

Daisy Chain — Linkage of primary and secondary readers allowing data to be relayed up to the host via auxiliary port connections.

Decode — A **Good Read**. The successful interpretation and output of the information encoded in a symbol.

Default — Restores **ROM** or flash settings and initializes serial commands.

Delimited — A delimited command or field is bracketed by predefined characters.

Decode Rate — The number of good reads per second ahieved by a reader.

Darkfield 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.

Depth-of-Field — The in-focus range of a reader. 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-to-Analog Converter (DAC) — A **VLSI** circuit used to convert digitally processed images to analog for display on a monitor.

Digital Signal Processor (DSP) — A **VLSI** chip designed for ultra-high-speed arithmetic processing.

Discrete I/O — Inputs and outputs characterized by discrete signal transitions from one voltage level to another so that digital switching can occur.

Direct Memory Access (DMA) — A capability provided by some computer bus architectures that allows data to be sent directly to memory from an attached device.

DSP — See **Digital Signal Processor**.

EPROM — See **Erasable Programmable Read-Only Memory**.

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 device containing programmable interconnects and logic components.

Firmware — Software hard-coded in non-volatile memory (**ROM**), and closely tied to specific pieces of hardware.

Fixed Symbol Length — Increases data integrity by ensuring that only a symbol length will be accepted.

Focal Distance — 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.

FPGA — See **Field-Programmable Gate Array**.

Full Duplex — A communications system in which signals can travel simultaneously between devices.

Good Read — A decode. The successful scanning and decoding of the information encoded in a bar code symbol.

Half Duplex — A communications system in which signals can travel between devices in both directions, but not simultaneously.

Host — A computer, **PLC**, or other device that is used to execute commands and process data and discrete signals.

Image Sensor — A device that converts a visual image to an electrical signal; a **CCD**, for example.

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.

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.

Multidrop — A communications protocol for networking two or more readers or other devices with a concentrator (or controller) and characterized by the use of individual device addresses and the RS-485 standard.

Normally Closed — A discrete output state that is only active when open.

Normally Open — A discrete output state that is only active when closed.

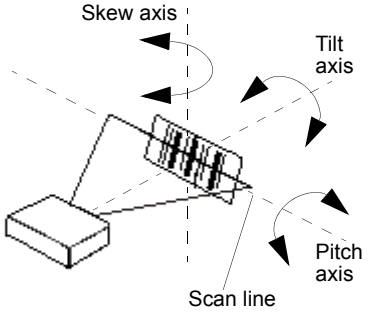
Object Plane — An imaginary plane in the field of view, focused by a reader's optical system at the corresponding image plane on the sensor.

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 **1** 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.



PLC — See **Programmable Logic Controller**.

Port — Logical circuit for data entry and exit. (One or more ports may be included within a single connector.)

Programmable Logic Controller (PLC) — An electronic device used in industrial automation environments such as factory assembly lines and automotive manufacturing facilities.

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.

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.

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 symbol is printed, stamped, or etched.

Symbol Transitions — The transition of bars and spaces on a symbol, used to detect the presence of a symbol on an object.

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.

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.

VLSI — See **Very Large-Scale Integration**.