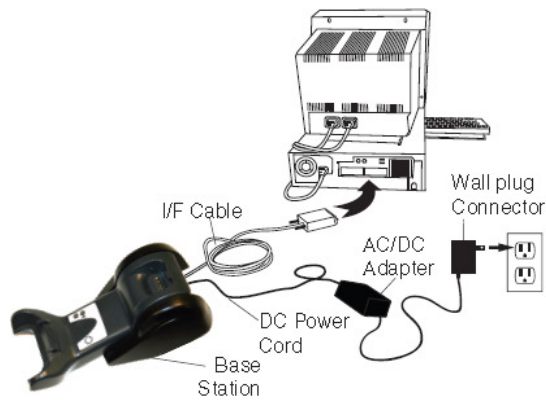


This document describes the serial image capture protocol between host and device.

Some Wasp 2D imagers (scanners) have the ability to capture images and send the image data to the host via RS232 or USB-COM modes (this is not the usual USB mode).



a) Serial Port mode (RS232 connector on host)



OR

b) USB-COM mode (USB connector on host)



Below you can find the basic settings to configure the scanner.

**1) Set up the scanner for RS232** Interface 115200 baud:



OR

**2) Set up the scanner for USB-COM :**

When setting up the scanner in USB-COM, a virtual USB-COM driver must be downloaded and installed from the following link:

<http://support.waspbarcode.com/kb/articles/1094>

Scan to enable USB-COM Interface :



### 3) Defining the image format

By default, the image format sent from the scanner to the host is Jpeg. This is the smallest file format of all and so will be the fastest to download. Other options are BMP, TIFF & JPG2000



JPEG (default)



TIFF



BMP



JPG2000

### 4) Request Image

To request a picture from the scanner, a command needs to be sent to the serial port on where the device is connected.

Multiple commands exist depending of how the reader reacts to the image taking command.

- a) set the reader to transmit an image after a trigger press : command = x018000000000{CR}
- b) acquire the image by using only the remote command : command = x008000000000{CR}

After the command is sent from the host to the scanner, a response answer string comes from the scanner. The format is like : \$i00000586360334{CR}

You can use the freeware tool ComtestSerial from Microridge to simulate this behaviour.

<http://www.microridge.com/ftp/comtestserial/>

In the below picture you see a print screen from the Serial test Program.

The host is sending : x008000000000{CR} and the scanner is answering \$i0100004c700357{CR}

The format of the answer string is always like :

\$ixyyyyyyyyzzcc{CR}

Where the description of the fields is :

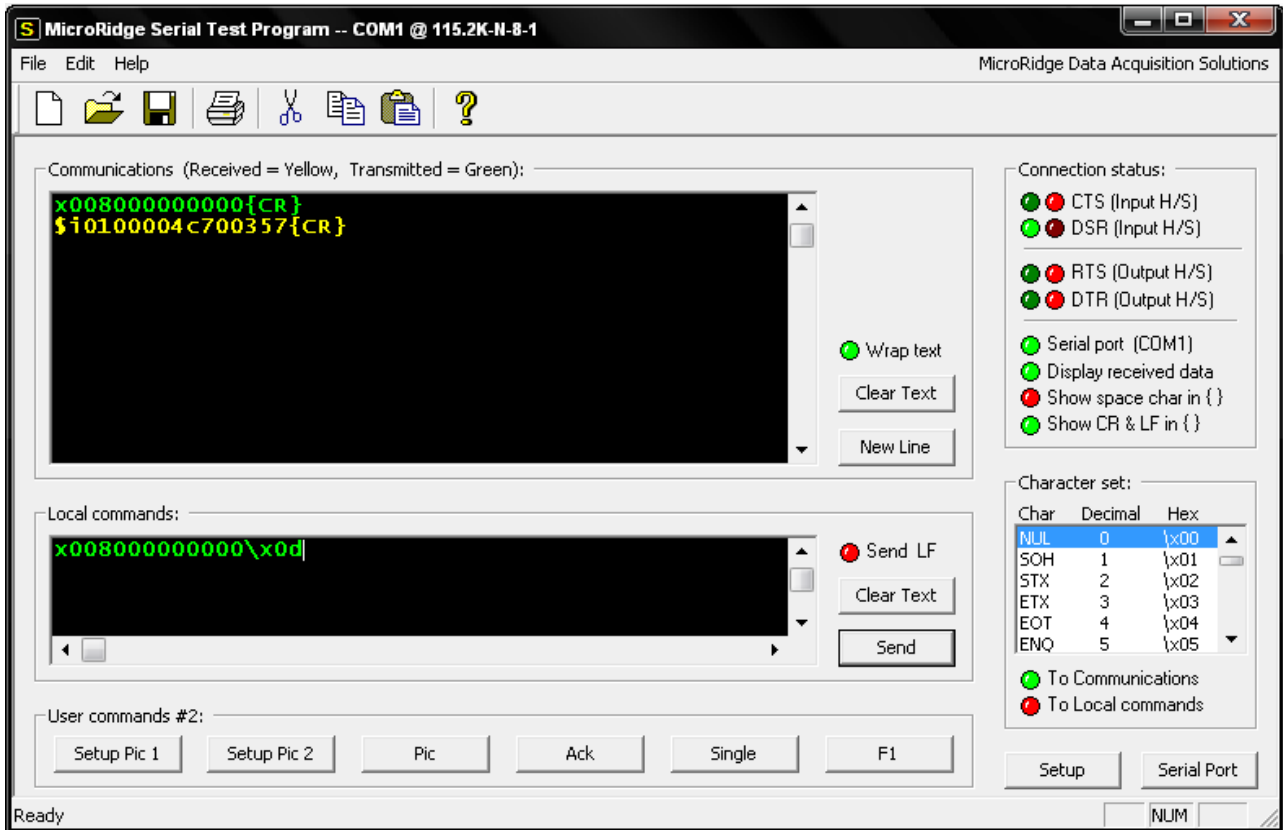
\$i = format header from a request picture answer. This is always \$i

xx = 2 digit Image format type (where : xx = 01 for JPEG , xx = 02 for JPEG2000 , xx = 03 for TIFF , xx = 00 for BMP)

yyyyyyyy = 8 digit hex value of the image size the scanner will send.

zz = 03 (fix)

cc = checksum (CRC 16)



After sending the image request answer to the host, the scanner will start sending the image data. Depending on the Interface selection of the scanner, 2 different situations can occur.

When working in RS232, the image data is sent from the scanner in X-modem 128 bytes protocol. When working in USB-COM, the image data is sent in a continuous stream without any handshaking.

RS232 mode flow :

Scanner		Host
	←	x018000000000{CR}
\$i010000249a0359{CR}	→	
	←	NAK
SOH 01 FE Data[128] CSUM	→	
	←	ACK
SOH 02 FD Data[128] CSUM	→	
	←	ACK
SOH 03 FC Data[128] CSUM	→	
	←	ACK
<i>..... multiple blocks will be sent .....</i>		
SOH xx FA Data[xx] CPMEOF[xx] CSUM	→	
	←	ACK
EOT	→	
	←	ACK

More information on the X-modem protocol can be found on :  
<http://www.techheap.com/communication/modems/xmodem.html>

In the Printscreen below, you will see that the host is sending x008000000000{CR} , the receives \$i0100003882032e{CR}, and starts the x-modem protocol by sending {NAK}

The screenshot shows the MicroRidge Serial Test Program interface. The title bar reads "MicroRidge Serial Test Program -- COM1 @ 115.2K-N-8-1". The menu bar includes "File", "Edit", and "Help". The toolbar contains icons for file operations and help. The main window is divided into several sections:

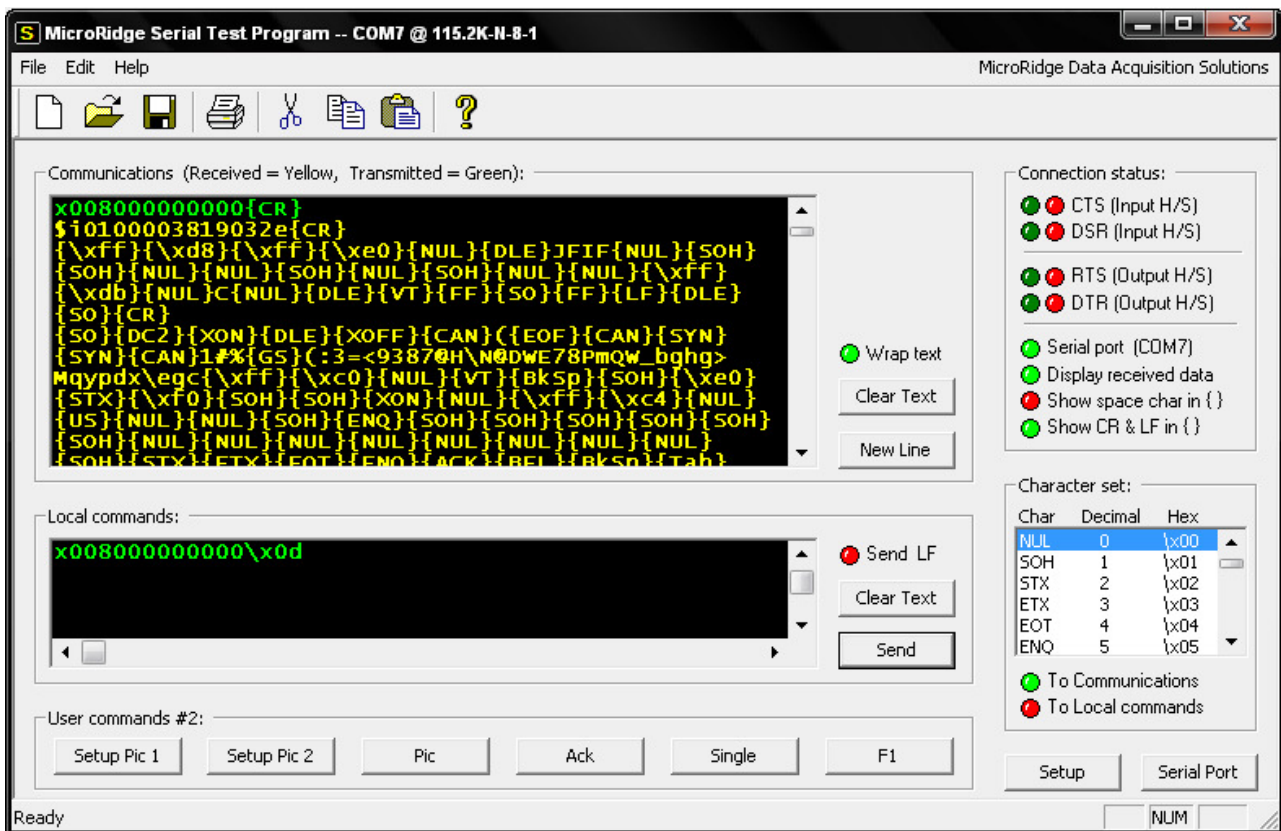
- Communications (Received = Yellow, Transmitted = Green):** A large text area displaying a stream of hexadecimal and control characters. The text is yellow, indicating it was received. The characters include: `x008000000000{CR}`, `$i0100003882032e{CR}`, `{NAK}{SOH}{SOH}{\xfe}{\xff}{\xd8}{\xff}{\xe0}`, `{NUL}{DLE}JFIF{NUL}{SOH}{SOH}{NUL}{NUL}{SOH}{NUL}`, `{SOH}{NUL}{NUL}{\xff}{\xdb}{NUL}c{NUL}{DLE}{VT}`, `{FF}{SO}{FF}{LF}{DLE}{SO}{CR}`, `{SO}{DC2}{XON}{DLE}{XOFF}{CAN}{EOF}{CAN}{SYN}`, `{SYN}{CAN}1#%{GS}(:3=<93870H\N0DWE78PmQW_bghq>`, `Mqypdx\egc{\xff}{\xc0}{NUL}{VT}{BkSp}{SOH}{\xe0}`, `{STX}{\xf0}{SOH}{SOH}{XON}{NUL}{\xff}{\xc4}{NUL}`, `{US}{NUL}{NUL}{SOH}{ENQ}{SOH}{SOH}{SOH}{SOH}`, `{SOH}{NUL}{NUL}{NUL}{NUL}{NUL}{NUL}{NUL}`, `{SOH}{STX}{ETX}{EOT}{RS}{SOH}{SOH}{\xfe}{\xff}`.
- Local commands:** A text area containing `x008000000000\x0d`. A red "Send LF" button is active.
- User commands #2:** A row of buttons: "Setup Pic 1", "Setup Pic 2", "Pic", "Ack", "Single", and "F1".
- Connection status:** A panel with status indicators for CTS (Input H/S), DSR (Input H/S), RTS (Output H/S), and DTR (Output H/S). It also shows "Serial port (COM1)", "Display received data" (checked), "Show space char in {}" (unchecked), and "Show CR & LF in {}" (checked).
- Character set:** A table with columns "Char", "Decimal", and "Hex". The "NUL" character (0, \x00) is selected.

At the bottom left, the status is "Ready". At the bottom right, there is a "Serial Port" dropdown menu currently set to "NUM".

USB-COM mode flow :

Scanner		Host
	←	x018000000000{CR}
\$i010000249a0359{CR}	→	
Image Data (number of bytes are known from the size information field in the above answer.	→	

Printscreen of JPEG image data received in USB-COM



As you see, the data is starting with {\xff}{xd8} and ending with {\xff}{xd9}. In fact these are the JPEG data headers and trailers as described in the JPEG format description :

<http://en.wikipedia.org/wiki/JPEG>

## 5) Additional information

The above described commands to capture a picture can be extended with following information :

If you want to download the images with different brightness, you can set the available options on the basic command:

```
x0nnqaabbccdd<CR>
```

- nn specifies the command type:

“08” -> HOST\_CMD\_CAPTURE\_NO\_HANDSHAKE

“10” -> HOST\_CMD\_CAPTURE\_ON\_TRIGGER

“11” -> HOST\_CMD\_MULTIPLE\_CAPTURE\_ON\_TRIGGER

“18” -> HOST\_CMD\_CAPTURE\_ON\_TRIGGER\_NO\_HANDSHAKE

“19” -> HOST\_CMD\_MULTIPLE\_CAPTURE\_ON\_TRIGGER\_NO\_HANDSHAKE

“40” -> HOST\_CMD\_ABORT\_IMAGE\_CAPTURE

- q specifies brightness and contrast. If q is different from ‘1’, the configured brightness and contrast will be applied. In this case all the remaining fields aabbccdd don’t matter.

If q = ‘1’ ASCII(0x31) then image brightness and contrast are specified by the following fields:

- aa = brightness value, range “00” to “64” (example “2A” for brightness 42%)

- bb = contrast value, range “00” to “64”

- cc = brightness direction: “00” to increase it; “01” means decrease.

- dd = contrast direction: “00” to increase it; “01” to decrease.