



## **FrameWork Firmware 2.6 Command List**

© Copyright 1996-2003 DVT Corp.  
All rights reserved.



# FrameWork 2.6 Firmware Commands

|                                                                   |           |
|-------------------------------------------------------------------|-----------|
| <b>SAMPLE TERMINAL COMMAND SESSION.....</b>                       | <b>1</b>  |
| <b>BINARY PACKET FORMAT .....</b>                                 | <b>3</b>  |
| <b>ACCESS CONTROL COMMANDS.....</b>                               | <b>4</b>  |
| <b>BACKGROUND TRANSFER COMMANDS .....</b>                         | <b>7</b>  |
| BACKGROUND GRAPH PACKET FORMAT AND SEQUENCE .....                 | 11        |
| BACKGROUND PERFORMANCE DATA FORMAT AND SEQUENCE .....             | 13        |
| IMAGE DATA FORMAT AND SEQUENCE.....                               | 14        |
| <b>COMMUNICATIONS COMMANDS.....</b>                               | <b>15</b> |
| <b>DIGITAL I/O COMMANDS.....</b>                                  | <b>17</b> |
| <b>FDA LOGGER COMMANDS .....</b>                                  | <b>21</b> |
| <b>IMAGE MANAGEMENT COMMANDS.....</b>                             | <b>22</b> |
| <b>SCRIPT VM COMMANDS .....</b>                                   | <b>28</b> |
| <b>NETWORK COMMANDS .....</b>                                     | <b>29</b> |
| <b>PRODUCT MANAGEMENT COMMANDS.....</b>                           | <b>30</b> |
| <b>REGISTER COMMANDS .....</b>                                    | <b>36</b> |
| <b>REAL-TIME CLOCK INTERFACE.....</b>                             | <b>41</b> |
| <b>SENSOR MANAGEMENT COMMANDS.....</b>                            | <b>37</b> |
| <b>TERMINAL MANAGEMENT COMMANDS.....</b>                          | <b>41</b> |
| <b>SYSTEM MANAGEMENT COMMANDS .....</b>                           | <b>47</b> |
| <b>SENSOR INPUT/OUTPUT SCHEME.....</b>                            | <b>51</b> |
| SPECIFIC INPUT/OUTPUT NUMBERS .....                               | 51        |
| SPECIFIC SOFTSENSOR RESULT CODES .....                            | 54        |
| <b>SETTING THRESHOLDS FOR ANY SENSOR.....</b>                     | <b>56</b> |
| COMMAND FORMAT FOR SETTING THRESHOLDS .....                       | 56        |
| <b>QUERY RESPONSE OF THRESHOLDS FOR ANY SENSOR.....</b>           | <b>57</b> |
| <b>SETTING COMMON PARAMETERS FOR ANY SENSOR .....</b>             | <b>58</b> |
| COMMAND FORMAT FOR DISPLAYING ACTIVE PLANE ON COLOR CAMERAS ..... | 58        |
| COMMAND FORMAT FOR SETTING MAX INSPECTION TIME.....               | 58        |
| COMMAND FORMAT FOR CHANGING DOMINANT COLOR THRESHOLD .....        | 58        |

|                                                                                     |           |
|-------------------------------------------------------------------------------------|-----------|
| COMMAND FORMAT FOR SETTING PROCESSING DOMAIN .....                                  | 58        |
| COMMAND FORMAT FOR SETTING LEARNED POSITION .....                                   | 58        |
| COMMAND FORMAT FOR SETTING POSITION REFERENCE .....                                 | 58        |
| <b>SETTING SHAPE FOR ANY SENSOR .....</b>                                           | <b>59</b> |
| COMMAND FORMAT FOR SETTING SENSOR SHAPE .....                                       | 59        |
| SPECIFIC SHAPES ASSOCIATED WITH EVERY SOFTSENSOR.....                               | 60        |
| <b>EDGE COUNTING SENSORS (MODULE ID: 1) .....</b>                                   | <b>64</b> |
| COMMAND FORMAT FOR SETTING EDGECOUNTING SENSOR PARAMETERS.....                      | 64        |
| EDGECOUNTING SENSOR QUERY RESPONSE .....                                            | 65        |
| EDGECOUNTING SENSOR OUTPUT QUERY RESPONSE .....                                     | 66        |
| <b>INTENSITY SENSORS (MODULE ID: 2).....</b>                                        | <b>67</b> |
| COMMAND FORMAT FOR SETTING INTENSITY REFERENCE SENSOR PARAMETERS .....              | 67        |
| INTENSITY REFERENCE SENSOR QUERY RESPONSE.....                                      | 68        |
| <b>POSITIONING SENSORS (MODULE ID: 3).....</b>                                      | <b>70</b> |
| COMMAND FORMAT FOR SETTING POSITIONING SENSOR PARAMETERS .....                      | 70        |
| POSITIONING SENSOR QUERY RESPONSE .....                                             | 72        |
| <b>PRECISIONMEASURE SENSORS (MODULE ID: 4).....</b>                                 | <b>74</b> |
| COMMAND FORMAT FOR SETTING PRECISIONMEASURE SENSOR PARAMETERS.....                  | 74        |
| PRECISIONMEASURE SENSOR QUERY RESPONSE .....                                        | 76        |
| <b>MATH SENSORS (MODULE ID: 5) .....</b>                                            | <b>78</b> |
| COMMAND FORMAT FOR SETTING MATH SENSOR PARAMETERS.....                              | 78        |
| MATH SENSOR QUERY RESPONSE .....                                                    | 80        |
| <b>MATH (COORDINATE SYSTEM/TYPE==10) SENSOR QUERY: .....</b>                        | <b>82</b> |
| COMMAND FORMAT FOR SETTING MATH(COORDINATE SYSTEM/TYPE==10) SENSOR PARAMETERS ..... | 82        |
| <b>DATA MATRIX SOFTSENSOR (MODULE ID: 6).....</b>                                   | <b>84</b> |
| COMMAND FORMAT FOR SETTING DATA MATRIX SENSOR PARAMETERS .....                      | 84        |
| DATA MATRIX SENSOR QUERY RESPONSE.....                                              | 86        |
| <b>BLOB GENERATION SOFTSENSOR (MODULE ID: 7) .....</b>                              | <b>87</b> |
| COMMAND FORMAT FOR SETTING BLOB GENERATION SENSOR PARAMETERS .....                  | 87        |
| BLOB GENERATION SENSOR QUERY RESPONSE .....                                         | 88        |
| <b>BLOB SELECTION SOFTSENSOR (MODULE ID: 8) .....</b>                               | <b>90</b> |
| COMMAND FORMAT FOR SETTING BLOB SELECTION SENSOR PARAMETERS .....                   | 90        |
| BLOB SELECTION SENSOR QUERY RESPONSE .....                                          | 91        |
| <b>SCRIPTING SOFTSENSOR (MODULE ID: 9) .....</b>                                    | <b>92</b> |
| COMMAND FORMAT FOR SETTING SCRIPTING SENSOR PARAMETERS .....                        | 92        |
| SCRIPTING SENSOR QUERY RESPONSE .....                                               | 93        |
| <b>OCR SOFTSENSORS (MODULE ID: 13) .....</b>                                        | <b>95</b> |
| COMMAND FORMAT FOR SETTING OCR SENSOR PARAMETERS .....                              | 95        |
| OCR SENSOR QUERY RESPONSE .....                                                     | 96        |
| <b>VERICODE SOFTSENSORS (MODULE ID: 14).....</b>                                    | <b>99</b> |
| COMMAND FORMAT FOR SETTING VERICODE SENSOR PARAMETERS .....                         | 99        |
| VERICODE SENSOR QUERY RESPONSE .....                                                | 100       |

|                                                                                       |            |
|---------------------------------------------------------------------------------------|------------|
| <b>SMARTLINK SOFTSENSORS (MODULE ID: 15).....</b>                                     | <b>102</b> |
| COMMAND FORMAT FOR SETTING SMARTLINK SENSOR PARAMETERS .....                          | 102        |
| SMARTLINK SENSOR QUERY RESPONSE.....                                                  | 102        |
| <b>SNOWFLAKE SOFTSENSORS (MODULE ID: 16) .....</b>                                    | <b>104</b> |
| COMMAND FORMAT FOR SETTING SNOWFLAKE SENSOR PARAMETERS.....                           | 104        |
| SNOWFLAKE SENSOR QUERY RESPONSE .....                                                 | 104        |
| <b>COLOR SOFTSENSORS (MODULE ID: 17).....</b>                                         | <b>106</b> |
| COMMAND FORMAT FOR SETTING PIXEL COUNTING AND COLOR MONITORING SENSOR PARAMETERS..... | 106        |
| COLOR SENSOR QUERY RESPONSE.....                                                      | 108        |
| <b>SEGMENTATION SOFTSENSOR (MODULE ID: 18).....</b>                                   | <b>109</b> |
| COMMAND FORMAT FOR SETTING SEGMENTATION SENSOR PARAMETERS .....                       | 109        |
| SEGMENTATION SENSOR QUERY RESPONSE.....                                               | 109        |
| <b>SPECTROGRAPH SOFTSENSOR (MODULE ID: 19) .....</b>                                  | <b>110</b> |
| COMMAND FORMAT FOR SETTING SPECTROGRAPH SENSOR PARAMETERS .....                       | 110        |
| SPECTROGRAPH SENSOR QUERY RESPONSE .....                                              | 110        |



## Sample Terminal Command Session

Framework commands and responses can contain strings of ASCII characters and numerical information. Numbers can be in one of three formats - Decimal, Hexadecimal, or Binary. All commands and responses are prefixed with a character identifying the command format. These prefix characters are:

| Format      | Command | Reply | End Reply |
|-------------|---------|-------|-----------|
| Decimal     | #       | \$    | %         |
| Hexadecimal | *       | (     | )         |
| Binary      | 0xAA    | 0xA8  | 0xA9      |

In Decimal mode, all numerical parameters are separated by a space character. All commands are terminated with a CR. Each reply sequence is completed with an EndReply with a 0 or 1 representing success or error, respectively, and a then followed by a ? or ! for success or error.

To bring up a FrameWork terminal, run FrameWork from within Windows and connect to a DVT system running FrameWork 2.6. From the menu select **Edit, Terminal Window...** or press **Control Alt T** to open a terminal window. Then use the commands shown below to configure your specific sensors. Commands entered by the user are in **bold** and responses are in *Italics*.

Example: System has two products defined. Product 'Clip' (id=1) is selected as the inspection product and the power on inspection product.

| <b>Command/Response</b>  | <b>Description</b>                                                                                        |
|--------------------------|-----------------------------------------------------------------------------------------------------------|
| <b>#PQ</b>               | Query system for products                                                                                 |
| <i>\$ -1 1 1</i>         | No selected Product, Product 1 set as power-on and inspection product. Product 0 is named 'Test Product'. |
| <i>\$ 0 Test Product</i> |                                                                                                           |
| <i>\$ 1 Clip</i>         | Product 1 is named 'Clip'.                                                                                |
| <i>% 0</i>               | Command completed without error.                                                                          |
| <i>?</i>                 | Command completed without error.                                                                          |
| <b>#Pq 2</b>             | Query product 2 for pass/fail stats.                                                                      |
| <i>% 1</i>               | Error performing command.                                                                                 |
| <i>!</i>                 | Error performing command.                                                                                 |
| <b>#Pq 1</b>             | Query product 1 for pass/fail stats.                                                                      |
| <i>\$ 100 2 3 -1</i>     | Product 1 has performed 100 Pass, 2 Warn, 3 Fail inspections, and the last inspection was Pass.           |
| <i>% 0</i>               |                                                                                                           |
| <i>?</i>                 | Command completed without error.                                                                          |
| <b>#PS 1</b>             | Select Product 1                                                                                          |
| <i>% 0</i>               | Command completed without error.                                                                          |
| <i>?</i>                 | Command completed without error.                                                                          |
| <b>#SQ</b>               | Query selected product ('Clip') for defined sensors.                                                      |
| <i>\$ 2 1 Sensor 2</i>   | 'Sensor 2' has an ID of 2 and is a type 1 sensor.                                                         |

\$ 5 2 *Test Path*

% 0

?

**#SPO 2 48**

% 0

?

**#SPO 5 16**

% 0

?

**#Sq 0**

\$ 0 1 *Clip Fold*

\$ 101 2 2 0

%0

?

‘Test Path’ has an ID of 5 and is a type 2 sensor.

Command completed without error.

Command completed without error.

Set ‘Sensor 2’ (ID=2) to activate user outputs OUT5 and  
OUT6 on failure. (16(OUT5)+32(OUT6) = 48)

Command completed without error.

Command completed without error.

Set ‘Test Path’ (ID=5) to activate user output OUT5 on  
failure.

Command completed without error.

Command completed without error.

Query Sensor 0 of selected product for pass/fail stats.

Sensor 0 id is 0, is active, and is named ‘Clip Fold’.

Sensor 0 has performed 101 Pass, 2 Warn, and 3 Fail  
inspections, and the last inspection result was Pass

Command completed without error.

## Binary Packet Format

With decimal or hexadecimal packet formats, the end of the packet can be easily determined by parsing until a carriage return is received. When the system is sending data in binary format, additional information is sent in order to allow the system to conveniently separate individual packets in the stream. After the command type is sent (0xA8, 0xA9, 0xAA), a 16 bit unsigned packet length is sent and then the data for the packet. The current system limits a single packet to 2048 bytes of data.

In binary mode, the type of the data must be adhered to explicitly in order to correctly read and write to the system. The descriptions for the commands give the explicit type for each data element.

The data in the binary packets is stored in big endian order. With a computer system that operates in little endian mode, such as an Intel PC, each multibyte entity must have the byte order reversed.

In order to reliably decode the binary information, a set of functions should be written to parse the individual packets from the information stream. A second set of functions should be written in order to read data of the appropriate types (UINT8, UINT16, UINT32, INT8, INT16, INT32, FLOAT32, FLOAT64, String) from the packet.

## Access Control Commands

Access control commands allow the user interface to query and set parameters used to control/limit the access of unauthorized users.

**AC** <User Name : string 20> '\t' <Current Password : string 20> '\t'  
<New Password : string 20> '\t'

Change the password for the specified user. Note that the passwords are not in plain text.

Access Restrictions: ALLOW\_ADMINISTRATION

Response:

<Operation Result : UINT32> <Flash Result : UINT8>

Operation result

AM\_NO\_SUCH\_USER (1): user with specified user name does not exist.

AM\_WRONG\_PASSWORD (2): Specified 'User's current password' does not match the one in database.

AM\_ACCESS\_GRANTED (0): Password successfully changed.

Flash result: UINT8

1 (TRUE ): changes were successfully saved to flash.

0 (FALSE): could not save to flash – may need to reclaim.

**AL** <User Name : string 20> '\t' <Password : string 20> '\t'  
<<Reason : string 200> >

Login the user for the terminal. The password is not in plain text. If successful this function will set UserName, UserCaps, and FullName fields in the terminal to match those of the current user. Login procedure will try to generate the Login event that will include Reason for login if it is provided.

Note: if logging is enabled but is not working (logging service is down) only administrator is allowed to log in.

Response on success:

<Access rights : UINT32> < User full name : string>

No response on failure.

**A1** <User Name : string 20> '\t' <Password : string 20>

Login the user using plain text for the password.

Note: if logging is enabled but is not working (logging service is down) only administrator is allowed to log in.

Response on success:

<Access rights : UINT32> < User full name : string>

No response on failure.

**A0**

Log out the current user. This sets the user back to default.

Response:

<Access rights : UINT32> < User full name : string>

**AQ**

Query list of users.

Response:

<Type : UINT8> <Rights : UINT32> <User name : string 20>  
'\t' <Full name : string 39> '\t'  
<Password : string 20>  
(repeat for each user)

The passwords are not in plain text.

**ASI** <Number Records : UINT32>

Initialize transfer of a set of user accounts. The number of records must be at least 2 in order to include the Administrator and the Default user.

Access Restrictions: ALLOW\_ADMINISTRATION

**ASA** <<Type : UINT8> <Rights : UINT32> <User name : string 20> '\t'  
<Full name : string 39> '\t' <Password : string 20>

Add record to set of user accounts.

Access Restrictions: ALLOW\_ADMINISTRATION

**ASC**

Cancel the transfer of user accounts.

Access Restrictions: ALLOW\_ADMINISTRATION

**ATG**

Get the session timeout value.

Response:

*<Timeout value : INT32>*

**ATS** *<Timeout value : INT32>*

Set session timeout value.

Access Restrictions: ALLOW\_ADMINISTRATION

Note: Session timeout is not currently being used by firmware.

**AU**

Get the current user that is logged in.

Response:

*<User name : string> '\t' <User full name : string> '\t'  
<Access rights : UINT32>*

## Background Transfer Commands

The background channel is designed to allow the user to configure a set of information that is needed along with the set of events when this should be sent and then have the firmware automatically keep this information current. The structure of the transfers is designed for a user interface application. The DVT user interface creates some of the graphs based on a single set of data and other graphs from the data accumulated over the history of the data transfers.

The data packets sent by the firmware as part of the background transfers is formatted as a command packet instead of a reply packet. This allows the processing routine to easily differentiate between replies from commands and background transfers.

The standard method of operation on the background channel is as follows:

1. Create connection to SmartImage Sensor.
2. Send the command to change to background channel mode of operation (BS).
3. Send the commands to setup and configure the types of data to transfer.
4. Start the playback of information (B+).
5. Perform any changes during the course of the operation.
6. Stop the playback of information (B-).
7. Close the background channel operation (TT).
8. Close the connection.

After the transmission of an image the background channel can be configured to either keep the image locked or to unlock it. This is selected when the operation is started using the BS command. If the image will not be needed for other information then the system should be configured to not hold the image in order to reduce the resource usage.

### **B+**

Start background transfers. If inspections are not enabled, then user mode pictures will be acquired based on sync mode.

### **B-**

Stop background transfers.

**BAG** <Product ID : INT16> <Sensor ID : INT16> <Graph Type : INT8>

Add background graph transfers to list of items to transfer. *Sensor ID* = -1 for all sensors.

**BAH** <Product ID : INT16> <Sensor ID : INT16> <History Type : INT8>

Add background performance/history transfers to list of items to transfer. *Sensor ID* = -1 for all sensors.

**BAI** <Win.P1.X : INT16> <Win.P1.Y : INT16> <Win.P2.X : INT16>  
<Win.P2.Y : INT16> <RowSkip : UINT8> <ColumnSkip : UINT8>  
<Finish : UINT8>

Add background image transfers to list of items to transfer. The window (Win) is the area to be transferred. RowSkip and ColumnSkip allow lower resolution images to be sent by skipping the specified number of pixels. A skip count of 0 sends the entire image in the window. The Finish flag indicates that a full resolution image without compression should be sent when other information does not need to be sent.

**BDA**

Remove all background transfers from list of items to transfer.

**BDG** <Product ID : INT16> <Sensor ID : INT16> <History Type : INT8>

Remove a background graph from list of items to transfer. The specification must match that used to add the transfer.

**BDH** <Product ID : INT16> <Sensor ID : INT16> <History Type : INT8>

Remove a background performance/history transfer from list of items to transfer. The specification must match that used to add the transfer.

**BDI**

Remove background image transfers from list of items to transfer.

**BF** <Finish : INT8>

The Finish flag indicates that a full resolution image without compression should be sent when other information does not need to be sent. This is the same Finish flag from the BAI command.

**BG** <Image ID : INT32> <Product ID : INT16> <Sensor ID : INT16>  
<Graph Type : INT8>

Send a graph on the background channel. The ImageID can be -2 for to indicate the newest image, -1 for the learn image, or a positive number to indicate a specific image. Transfers must be stopped for this command to work. This command is not used by the FrameWork user interface.

**BH**

Send previously configured performance information for products and sensors. Transfers must be stopped for this command to work.

**BI** <Image ID: INT32> <Win.P1.X : INT16> <Win.P1.Y : INT16>  
<Win.P2.X : INT16> <Win.P2.Y : INT16>

Send image on the background channel. The ImageID can be -2 to indicate the newest image, -1 for the learn image, or a positive number to indicate a specific image. This command is not used by the FrameWork user interface.

**BM** <autodisable : INT8> <mode : INT8> <priority low : INT8>  
<priority high : INT8> <Select : INT8>

Set background transfer mode. When *autodisable* is 1, the background channel will auto-disable after each transfer. When *mode* is 0, the background channel automatically changes to the product ID of the inspected product. When *mode* is 1, the image must match the product ID in order to be considered for transfer. The priority range indicates the inspection result range that an image must have had in order to be considered for transfer. *Select* is an additional image type of selection.

|            |                                                         |
|------------|---------------------------------------------------------|
| Select = 0 | All Images, priority -128 to 127                        |
| Select = 1 | Passing Only, priority -128 to -1                       |
| Select = 2 | Warning and Failed Only, priority 0 to 127              |
| Select = 3 | Failed Only, priority 15 to 127                         |
| Select = 4 | All Image but Stop on Next Failed, priority -128 to 127 |
| Select = 5 | Do not skip over images                                 |

**Bm** <100 : INT32> <1 : INT32> <300 : INT32> <0 :INT32>

Advanced mode selection.

**BP** <*Product ID* : INT16>

Select *Product ID* as the selected background product for image marking/evaluation.

**BS** <<*HoldImage* : INT8>>

Start a thread for background transfers. If *Hold Image* is 0 then the image is released after sending. The default value is 1 if not included. If the receiving program is not designed to request more information on this image after being received then *Hold Image* should be 0 to reduce the resource usage.

Response:

<Connection Count : UINT8><Terminal ID : INT32>

The *Connection Count* is the number of active background channels for the system. The *Terminal ID* is a unique identifier for this terminal and is used with the TL command.

**BT** <*Image ID* : INT32>

Send configured graph and history data on the background channel. The ImageID can be -2 to indicate the newest image, -1 for the learn image, or a positive number to indicate a specific image.

**BY** <*Mark Image* : INT16>

Select the type of marking the sensors should perform on the image.

- 1 No marking
- 2 All sensors with the active flag true
- 3 All sensors with the active flag false
- 4 All sensors
- >-1 Only the sensor with this ID

**BZ**

Turn off Active domain display for all sensors.

## ***Background Graph Packet Format and Sequence***

Graph data is normally sent in combination with the image data. It can be explicitly requested by using the 'BT' command. The format and sequence of the data is indicated below.

Header Packet:

*<(packet type) 0 : UINT8> <Image ID : INT32> <Time Acq : INT32>  
<Product ID : INT16>*

Graph Header Packet:

*<(packet type) 32 : UINT8> <Image ID : INT32>  
<Product ID : INT16> <Sensor ID : INT16> <Graph Type : INT8>  
<Data Format : INT8> <Data Length : INT32>  
<Graph Specific Data>*

Graph Data Packet:

*<(packet type) 34 : UINT8> <Graph Data>*

Graph End Packet:

*<(packet type) 33 : UINT8> <Image ID : INT32>  
<Product ID : INT16> <Sensor ID : INT16> <0 : INT8>  
... other graphs*

End Packet:

*<(packet type) 1 : UINT8> <Image ID : INT32>*

**Graph Type Info:**

| <b>Graph Type</b> | <b>Description</b>        | <b>Graph Specific Data</b>                                                                                                   |
|-------------------|---------------------------|------------------------------------------------------------------------------------------------------------------------------|
| 0                 | Pixel Intensity Graph     | <Threshold : INT32><br><Contrast : INT32><br><Min Intensity : INT32><br><Max Intensity : INT32><br><Number of Edges : INT32> |
| 1                 | Intensity Histogram Graph | <Threshold : INT32>                                                                                                          |
| 2                 | Pixel Gradient Graph      | <Threshold : INT32><br><Contrast : INT32><br><Min Intensity : INT32><br><Max Intensity : INT32><br><Number of Edges : INT32> |

**Data Format:**

| <b>Data Format</b> | <b>Description</b>       |
|--------------------|--------------------------|
| 3                  | 256 level grayscale data |

## Background Performance Data Format and Sequence

Performance data that is configured to be sent is output approximately every two seconds. The format and sequence of the data is indicated below.

Header Packet:

<(packet type) 2 : UINT8>

Perf/Hist. Packet:

<(packet type) 48 : UINT8> <Perf/Hist Type : UINT8>  
<Perf/Hist Data>

...

End Packet:

<(packet type) 3 : UINT8>

### Performance Packet Info:

| Entry Type | Entry Data                                                                                                                                         |
|------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| 0          | Product Pass/Warn/Fail counts<br><Product ID : INT16> <Fail Count : UINT32><br><Pass Count : UINT32> <Warn Count : UINT32>                         |
| 1          | Sensor PassWarn/Fail counts<br><Product ID : INT16> <Sensor ID : INT16><br><Fail Count : UINT32> <Pass Count : UINT32><br><Warn Count : UINT32>    |
| 2          | same as Sensor Specific 'Output' section of 'SQ' command<br><Product ID : INT16> <Sensor ID : INT16><br><data from SQ command>                     |
| 4          | same as 'Ii' command                                                                                                                               |
| 5          | same as 'YQ' command                                                                                                                               |
| 6          | Product Pass/Fail status<br><Product ID : INT16> <Result : INT8>                                                                                   |
| 7          | Sensor Pass/Fail status<br><Product ID : INT16> <Sensor ID : INT16> <Result1 : INT8><br><Result : INT8>                                            |
| 8          | I/O history data                                                                                                                                   |
| 9          | Current digital I/O state<br><input polarity : UINT16> <inputs : UINT16><br><raw inputs : UINT16> <output polarity : UINT16><br><outputs : UINT16> |

## Image Data Format and Sequence

The following information describes the uncompressed data format with the image marking performed directly on the image. The palette for an unmarked image should be 0 (black) to 255 (white). The palette for a marked image should be as follows:

| Entry | R   | G   | B   |
|-------|-----|-----|-----|
| 0     | 0   | 0   | 0   |
| 1     | 0   | 0   | 255 |
| 2     | 255 | 0   | 0   |
| 3     | 255 | 0   | 255 |
| 4     | 0   | 255 | 0   |
| 5     | 0   | 255 | 255 |
| 6     | 255 | 255 | 0   |
| 7     | 128 | 0   | 0   |
| 8     | 8   | 8   | 8   |
| 9     | 9   | 9   | 9   |
| 10    | 10  | 10  | 10  |
| ...   |     |     |     |
| 255   | 255 | 255 | 255 |

### Image Header Packets:

```
<(packet type) 16 : UINT8> <Image ID : INT32> <Win.P1.X : INT16>
<Win.P1.Y : INT16> <Win.P2.X : INT16> <Win.P2.Y : INT16>
<(Format) 0 : UINT8> <Plane : UINT16> <0 : UINT16>
<RowSkip : UINT8> <ColSkip : UINT8> <0 : UINT8> <0 : UINT8>
<AcqWin.P1.X : INT16> <AcqWin.P1.Y : INT16>
<AcqWin.P2.X : INT16> <AcqWin.P2.Y : INT16>
<8 : UINT8> <0 : UINT8> <0 : UINT8>
<Marked Image : UINT8> <Product Result : INT16>
<CCD Width : UINT16> <CCD Height : UINT16>

<(packet type) 20 : UINT8> <Image ID : INT32>
<Image Name : string>
```

### Image Data Packets:

```
<(packet type) 18 : UINT8> <Image ID : INT32> <image data>
... more image data packets
```

### End Packet:

```
<(packet type) 17 : UINT8> <Image ID : INT32>
```

### Complete Image Indicator Packet:

```
<(packet type) 21 : UINT8> <Image ID : INT32>
```

## Communications Commands

**Cb** <New Baud Rate : INT32>

**CB** <New Baud Rate : INT32>

Set a new baud rate.

Response:

<\r \n?: string>

**Ce** <+|- : char>

**CE** <+|- : char>

Enable (+) or disable (-) echo of data.

**Cq**

**CQ**

Query current communication configuration.

Response:

<Current Baud Rate : INT32> <Flow Control : INT8>

Flow Control:

0 = NONE

1 = XON/XOFF

2 = RTS/CTS

**Cx** <Data>

**CX** <Data>

Echo data back. This can be used in order to check for communication errors.

Response:

<\r\n?: string>

**Cn**

Query network communication parameters

Response:

```
<IP Address octet 1 : UINT8>
  <IP Address octet 2 : UINT8>
  <IP Address octet 3 : UINT8>
  <IP Address octet 4 : UINT8>
  <IP Mask octet 1 : UINT8>
  <IP Mask octet 2 : UINT8>
  <IP Mask octet 3 : UINT8>
  <IP Mask octet 4 : UINT8>
  <Gateway IP octet 1 : UINT8>
  <Gateway IP octet 2 : UINT8>
  <Gateway IP octet 3 : UINT8>
  <Gateway IP octet 4 : UINT8>
  <NetFlags : INT32>
```

**Cn** <IP Address octet 1 : UINT8> <IP Address octet 2 : UINT8>  
<IP Address octet 3 : UINT8> <IP Address octet 4 : UINT8>  
<IP Mask octet 1 : UINT8> <IP Mask octet 2 : UINT8>  
<IP Mask octet 3 : UINT8> <IP Mask octet 4 : UINT8>  
<Gateway IP octet 1 : UINT8> <Gateway IP octet 2 : UINT8>  
<Gateway IP octet 3 : UINT8> <Gateway IP octet 4 : UINT8>  
<NetFlags : INT32>

Set Network Communication parameters

Access Restrictions: ALLOW\_COMM\_CHANGES

## Digital I/O Commands

Digital I/O commands allow the configuration and mapping of virtual I/O bits to physical I/O, as well as the configuration of the physical I/O.

The physical I/O is different on the various DVT SmartImage Sensors. The tables below briefly describe the I/O for each DVT system.

### Model 630 / SmartReader

Pins 1-12: Can be selected as an input and/or an output. Pin 12 has special hardware that allows it to control very precisely the Strobe signal to a DVT strobe light, and should be configured for the Strobe function, if used. Pins 1-12 map to bits 0 – 11 of both the input and output masks.

Pass LED (Green) Bit 12

Trigger LED (Green) Bit 13

Fail LED (Green) Bit 14

Pass LED (Red) Bit 15

Trigger LED (Red) Bit 16

Fail LED (Red) Bit 17

### Legend 5XX / Intelligent Scanner / SpectroCam

Pins 1-8: Can be selected as an input and/or an output. Pins 1-8 map to bits 0-7 of both the input and output masks. Any pin can be used as a Strobe signal.

**DD** *<Input bit : UINT8> <Dactive : UINT16> <Dinactive : UINT16>*

Set Digital Input Debounce time (in ms)

Access Restrictions: ALLOW\_DIG\_IO\_DATA\_LINK

**DA** *<Start time : INT32>*

Get digital IO history

Response:

```
<Number of points : INT8> <' \r' >
{
  <user time : UINT32>
    <user physical inputs : UINT32>
    <user physical outputs : UINT32>
    <user virtual inputs : UINT32>
    <user virtual outputs (63 : 32) : UINT32>
    <user virtual outputs (31 : 0) : UINT32>
  } in groups of 4 separated by <'r'>
```

**DI** <*NormPolarity Bits* : UINT32> <*Inv Polarity Bits* : UINT32>

Sets input *NormPolarity Bits* to normal polarity and *InvPolarity Bits* to inverted polarity.

Access Restrictions: ALLOW\_DIG\_IO\_DATA\_LINK

**DM** <*Virtual Input bit* : UINT8> <*Mask* : UINT32>

Sets the virtual input mask for the specified bit.

Access Restrictions: ALLOW\_DIG\_IO\_DATA\_LINK

**DN** <*Virtual Output bit* : UINT8> <*Mask High* : UINT32><*Mask Low* : UINT32>

Sets the virtual output mask for the specified bit.

Access Restrictions: ALLOW\_DIG\_IO\_DATA\_LINK

**DO** <*NormPolarity Bits* : UINT32> <*Inv Polarity Bits* : UINT32>

Sets output *NormPolarity Bits* to normal polarity and *InvPolarity Bits* to inverted polarity.

Access Restrictions: ALLOW\_DIG\_IO\_DATA\_LINK

## DQ

Query general digital I/O info

Response:

```
<Number physical inputs : UINT8>
  <Number physical outputs : UINT8>
  <I/O Isolation : INT8>
  <physical input polarity : UINT32>
  <physical input state : UINT32>
  <physical raw input : UINT32>
  <virtual input state : UINT32>
  <physical output polarity : UINT32>
  <physical output state : UINT32>
  <virtual output state high : UINT32>
  <virtual output state low : UINT32>
<Bit 0 Active debounce time : UINT16>
  <Bit 0 Inactive debounce time : UINT16> ...
  <Bit N Active debounce time : UINT16>
  <Bit N Inactive debounce time : UINT16>
<virtual input mask 0 : UINT32> ...
  <virtual input mask 15 : UINT32>
<virtual output mask high 0 : UINT32>
  <virtual output mask low 0 : UINT32> ...
  <virtual output mask high 7 : UINT32>
  <virtual output mask low 7 : UINT32>
<virtual output mask high 8 : UINT32>
  <virtual output mask low 8 : UINT32> ...
  <virtual output mask high 15 : UINT32>
  <virtual output mask low 15 : UINT32>
```

## Dq

Query digital I/O states

Response:

```
<Number physical inputs : UINT8>
  <Number physical outputs : UINT8>
  <I/O Isolation : INT8>
  <physical input polarity : UINT32>
  <physical input state : UINT32>
  <physical raw input : UINT32>
  <virtual input state : UINT32>
  <physical output polarity : UINT32>
  <physical output state : UINT32>
  <virtual output state high : UINT32>
  <virtual output state low : UINT32>
```

**DS** <Active Bits : UINT32> <Inactive Bits : UINT32>

Set physical digital output bits *Active Bits* to the active state and *Inactive Bits* to the inactive state.

Access Restrictions: ALLOW\_DIG\_IO\_DATA\_LINK

## **Error Log Access**

**ec**

Clear the error log.

**Ed**

Dump contents of the error log.

## SmartLogger Commands

Commands to configure and control SmartLogger service.

### FGS

Query current state of the service

Response:

<System State : UINT8>

CONNECTING (1): attempting connection to the remote service

CONNECTED (2): connection to the remote service established

DISABLED (3): event logger is not enabled

### FGA

Query IP address and port number that event logger will try to connect to.

Response:

<Server IP address : UINT32> <Server Port : UINT16>

### FSS <New State : UINT8>

Set service state. A value of 0 disables the service and 1 enables the service.

Access Restrictions: ALLOW\_ADMINISTRATION

### FSA <Server IP Address : UINT32> <Server Port : UINT16>

Set IP address and port number that event logger will try to connect to.

Access Restrictions: ALLOW\_ADMINISTRATION

### FB <Password : string 21> '\t' <Save-to-flash flag : UINT8> <<Reason : string 200>>

Trigger an 'Explicit Backup' event and perform full system backup. The password is not in plain text. If the save-to-flash flag is 1 then the system state will be saved to flash.

Response:

<Result : UINT32>

### FC <Password : string 21> '\t' <<Reason : string 200>>

Custom Event.

## Image Management Commands

### I

Query version description.

Response:

*<version : string>*

### IC

Query image sensor info.

Response:

```
<select ID : INT16> <num installed : UINT8>
  <num acquisition/pipeline buffers : UINT8>
  <num buffers: UINT8>
  <ID1 : INT16> <#XPixels : INT16> <#YPixels : INT16>
  <PixelWidth : INT16> <PixelHeight : INT16>
  <Name : string 50>
  ...
  <IDn : INT16> <#XPixels : INT16> <#YPixels : INT16>
  <PixelWidth : INT16> <PixelHeight : INT16>
  <Name : string 50>
```

### Ic

Perform calibrate of the image sensor.

Access Restrictions: ALLOW\_IMAGE\_PARAMS

Response:

*<status : INT32>*  
Calibration status. (0=success, -1=fail)

### ID *<DeleteStrategy : UINT8>*

Set delete strategy from the saved image list.

- 0 : delete newest,
- 1 : delete oldest
- 2 : delete newest by priority
- 3 : delete oldest by priority

Access Restrictions: ALLOW\_IMAGE\_PARAMS

**IG** <Entry : INT16>

Get image sensor calibration value. These values are specific to the image sensor driver.

Response:

<Boot value : INT16>  
<Application value : INT16>  
<Current value : INT16>

**IG** <Entry : INT16><Value : INT16>

Set image sensor calibration value.

Access Restrictions: ALLOW\_IMAGE\_PARAMS

**Ii**

Query inspection statistics.

Response :

<Total Inspections : UINT32>  
<Missed Inspections : UINT32>  
<Acquisition Conflicts : UINT32>  
<Timeout Failures : UINT32>  
<Min Time in Capture Queue : UINT32>  
<Min Inspection Time : UINT32>  
<Mean Inspection Time : UINT32>  
<Max Inspection Time : UINT32>  
<Min Time between Acquisitions : UINT32>  
<Mean Time between Acquisitions : UINT32>

**Ii** <Inspection Stat : char>

Query specific inspection statistics value.

Response: (Inspection Stat)

a: <Min Time Between Acquisitions : UINT32>  
b: <Mean Time Between Acquisitions : UINT32>  
c: <Acquisition Conflicts : UINT32>  
f: <Timeout Failures : UINT32>  
i: <Min Inspection Time : UINT32>  
m: <Missed Inspections : UINT32>  
n: <Mean Inspection Time : UINT32>  
q: <Min Time in Capture Queue : UINT32>  
t: <Total Inspections : UINT32>  
x: <Max Inspection Time : UINT32>

## **Iiz**

Reset all inspection statistics.

## **Iiz** <Inspection Stat : char>

Reset specific inspection statistic.

Response: (Inspection Stat)

- a: Min Time Between Acquisitions (reset to 0x7FFFFFFF)
- b: Mean Time Between Acquisitions
- c: Acquisition Conflicts
- f: Timeout Failures
- i: Min Inspection Time (reset to 0x7FFFFFFF)
- m: Missed Inspections
- n: Mean Inspection Time
- q: Min Time in Capture Queue (reset to 0x7FFFFFFF)
- t: Total Inspections
- x: Max Inspection Time

## **Il** <PacketType : UINT8> <Specific Packet Data>

Load image as a set of packets. See the documented format in the Background Transfer Commands section.

Access Restrictions: ALLOW\_IMAGE\_PARAMS

Response: (Header Only)

<Assigned ImageID : INT32 >

## **IL**

Lock the Saved Image Queue. Save Image Queue is automatically unlocked if terminal terminates.

Access Restrictions: ALLOW\_IMAGE\_PARAMS

## **IP** <# system buffers : UINT8>

Set system image buffers. This is not saved to flash memory.

Access Restrictions: ALLOW\_IMAGE\_PARAMS

## **IQ**

Query image information

Response:

```
<delete strategy : UINT8>
<ID1 : INT32> <result : INT8>
  <product : INT16>
  <image handle : INT16>
  <First Failing/Warning Sensor ID : INT16>
  <First Failing /Warning Sensor Result Code: INT8>
  ...
<IDn : INT32> <result : INT8>
  <product : INT16>
  <image handle : INT16>
  <First Failing/Warning Sensor ID : INT16>
  <First Failing /Warning Sensor Result Code: INT8>
```

## **IQ** <image ID : INT32>

Query specific image information

Response:

```
<Result : INT8>
  <Capture Queue Time Stamp : UINT32>
  <Acquisition Time Stamp : UINT32>
  <Inspection Time Stamp : UINT32>
  <Saved Time Stamp : UINT32>
  <ProductID : INT16>
  <Lock Count : INT16>
  <Exposure Time : UINT32>
  <Antiblooming Enabled : UINT8>
  <IlluminationEnabled : UINT8>
  <X1 : INT16> <Y1 : INT16> <X2 : INT16> <Y2 : INT16>
  <AfterSyncDelay : UINT32>
  <AntiBloomingLevel : INT16>
  <Gain : INT16>
```

## **IR** <image ID : INT32>

Release the selected image from the Saved Queue.

Access Restrictions: ALLOW\_IMAGE\_PARAMS

## **IS** <image ID : INT32> <x1 : INT16> <y1 : INT16> <x2: INT16> <y2: INT16> <θ : UINT8> <rowskip : UINT8> <colskip : UINT8> <unused : INT8> <unused : UINT8>

Send image as a set of packets.

Response: see description in Background Transfer Commands section.

## Is

Send the newest image as a set of packets.

Response: see description in Background Transfer Commands section.

## IU

Unlock the Saved Image Queue

Access Restrictions: ALLOW\_IMAGE\_PARAMS

## Iv

Query FOV Balance data

Response:

```
<Is FOV Balance Calibrated : UINT8>
<Is FOV Balance Enabled : UINT8>
<Active Window X1 : UINT32>
<Active Window Y1 : UINT32>
<Active Window X2 : UINT32>
<Active Window Y2 : UINT32>
<Number of Saturated Pixels : INT32>
<Is Black Calibrated : UINT8>
```

**Iv** <Action : INT32> <Active Window X1 : UINT32>  
<Active Window Y1 : UINT32> <Active Window X2 : UINT32>  
<Active Window Y2 : UINT32>

FOV calibration commands. The value of *Action* indicates the type of operation to perform. If *Action* is not 3, the active window parameters can be any number since they are not used.

*Action* = -3 : Only valid for SpectroCam. First reads 42 FLOAT32 numbers for the universal white then saves the data to flash.

*Action* = -2 : Deletes the black calibration.

*Action* = -1 : Deletes the white calibration.

*Action* = 0 : Perform White Calibration.

*Action* = 1 : Enable FOV Balance.

*Action* = 2 : Disable FOV Balance.

*Action* = 3 : Set FOV active window.

*Action* = 4 : Save modified parameters to flash.

*Action* = 5 : Cancel all changes and load the previously saved data from flash.

*Action* = 6 : Perform black calibration.

Access Restrictions: ALLOW\_IMAGE\_PARAMS

Saving to flash allow requires: ALLOW\_SAVE\_TO\_FLASH

**Iw**

Query White balance gains for color camera

Response:

*<Red Gain : UINT32> <Green Gain : UINT32>  
<Blue Gain : UINT32>*

**Iw** *<Red Gain : UINT32> <Green Gain : UINT32> <Blue Gain : UINT32>  
<Save-to-Flash flag : UINT8>*

Sets White balance RGB gains of the color camera and saves to flash if the flag is not 0.

Access Restrictions: ALLOW\_IMAGE\_PARAMS

Saving to flash allow requires: ALLOW\_SAVE\_TO\_FLASH

**IW**

Query Spectral calibration points

Response:

*<Number of Points : UINT8>  
<Wave Length 1 : INT32><Pixel Position 1 : INT32> ...  
<Wave Length N : INT32><Pixel Position N : INT32>*

**IW** *<Action: INT32>*

Performs different operations as specified by the value of *Action*:

*Action = 0* : Set default calibration points.

*Action = 1* : Reads the points and performs calibration.

*<Number of Points : UINT8>  
<Wave Length 1 : INT32><Pixel Position 1 : INT32> ...  
<Wave Length 1 : INT32><Pixel Position 1 : INT32>*

*Action = 2* : Save data to flash.

*Action = 3* : Cancel and reload previously saved data from flash.

Access Restrictions: ALLOW\_IMAGE\_PARAMS

Saving to flash allow requires: ALLOW\_SAVE\_TO\_FLASH

## Script VM Commands

ZA – Add script – ALLOW\_BACKGND\_SCR  
ZD – Delete script – ALLOW\_BACKGND\_SCR  
ZL – List scripts  
ZI – debug, load from flash  
ZN – Start/stop script by name – ALLOW\_BACKGND\_SCR  
ZP – Set script parameter – ALLOW\_BACKGND\_SCR  
ZQ – query specific script  
ZR – Start/stop script – ALLOW\_BACKGND\_SCR  
ZS – Save script to flash – ALLOW\_BACKGND\_SCR

## **Network Commands**

NA – Set Address – ALLOW\_COMM\_CHANGES

## Product Management Commands

Note: If *Product ID* = -2 then the current default product is used

**PA** <*name* : string 50>

Add new product and auto name or set name to *name*. Product is created active.

Response:

<*ID* : INT16> <*name* : string 50>

Access Restrictions: ALLOW\_PRODUCT\_PARAMS

**PD** <*Product ID* : INT16>

Delete product and its sensors from RAM and Flash

Access Restrictions: ALLOW\_PRODUCT\_PARAMS and  
ALLOW\_SAVE\_TO\_FLASH

**PFD** <*Product ID* : INT16>

Delete product from flash

Access Restrictions: ALLOW\_SAVE\_TO\_FLASH

**PFS**

Save all products to flash

Access Restrictions: ALLOW\_SAVE\_TO\_FLASH

**PFS** <*Product ID* : INT16>

Save product to flash

Access Restrictions: ALLOW\_SAVE\_TO\_FLASH

**PI** <*Product ID*: INT16>

Select inspection product

Access Restrictions: ALLOW\_PRODUCT\_CHANGEOVER

**PL** <Sensor ID : INT16>

Learn all sensors (or optional selected sensor) for the selected product on the currently locked learn-image.

Access Restrictions: ALLOW\_SENSOR\_PARAMS and  
ALLOW\_SENSOR\_SHAPE

**PM** <Product ID: INT16>

Query general Data Link information.

Response:

< Product ID: INT16> <Transmit Rate : INT8>  
<Fail Character : char> < Placeholder Type : INT8>  
<Output Type : INT8>

|                   |   |                       |
|-------------------|---|-----------------------|
| Placeholder Type: | 0 | Spaces                |
|                   | 1 | Zeros                 |
| Output Type:      | 0 | Regular Output Format |
|                   | 1 | XML output format     |

**PM** <Product ID: INT16> <Data Link ID : INT8>

Query information on a specific Data Link string

Response:

<Product ID: INT16> <Data Link ID : INT8> <Function : INT8>  
<Priority Flag : INT8>  
<Total String Length : INT16>  
<Number of Characters in this line: INT16>  
<Data Link String : string>  
<Number of Characters in this line: INT16>  
<Data Link String : string>

|           |   |        |
|-----------|---|--------|
| Function: | 1 | Never  |
|           | 2 | Always |
|           | 3 | AND    |
|           | 4 | OR     |

**PN** <Product ID: INT16> <name : string 50>

Rename product

Access Restrictions: ALLOW\_PRODUCT\_PARAMS

**PP** <Product ID: INT16>

Set product parameters (-1 sets default parameters):

A <flag : UINT4>  
Set active product flag.  
Access Restrictions: ALLOW\_PRODUCT\_PARAMS

B <flag : UINT4>  
Set antiblooming enable flag.  
Access Restrictions: ALLOW\_IMAGE\_PARAMS

b <Antiblooming level : INT16>  
Set antiblooming level.  
Access Restrictions: ALLOW\_IMAGE\_PARAMS

d <Digitizing Rate : UINT16>  
Set image digitizing rate.  
Access Restrictions: ALLOW\_IMAGE\_PARAMS

D <time : UINT32>  
Set delay after trigger signal (microseconds).  
Access Restrictions: ALLOW\_IMAGE\_PARAMS

E <time : INT32>  
Set exposure time (microseconds, 0 .. 1 sec.)  
Access Restrictions: ALLOW\_IMAGE\_PARAMS

G <Imaging Gain : INT16>  
Set imaging gain (0-245).  
Access Restrictions: ALLOW\_IMAGE\_PARAMS

I <flag : UINT4>  
Set strobe light enable flag.  
Access Restrictions: ALLOW\_IMAGE\_PARAMS

M <Trans Rate : INT8> <Fail Char : char>  
<Terminate Type : INT8> <Output Type : INT8>  
Note: Output Type 0 = Standard Datalink Format,  
1 = XML  
Set general Data Link parameters.  
Access Restrictions: ALLOW\_DIG\_IO\_DATA\_LINK

m -- Parameters for specific Data Link string  
g <Data Link ID : INT8> <Function : INT8>  
<Priority : INT8>  
See note above for function values.  
Format for first string after Pmg command.  
s <DataLink ID : INT8> <Total String Length : INT16>  
<Chars in this command : INT16>  
<Data Link String : string>  
Format for second and following strings after Pmg command.  
s <DataLink ID : INT8> <Chars in this command : INT16>  
<DataLink String : string>  
Access Restrictions: ALLOW\_DIG\_IO\_DATA\_LINK

**O** <FailTime : INT16> <OutputPulseWidth : INT16>  
     <PostOutputDelay : INT16>  
     <ResConflictPulseWidth : INT16>  
     Set pulse widths and delays for an inspection.  
     Access Restrictions: ALLOW\_DIG\_IO\_DATA\_LINK  
**P** <all pass bits : UINT16>  
     Set bits that should go active if all sensors (with appropriate output bits set) pass.  
     Access Restriction: ALLOW\_DIG\_IO\_DATA\_LINK  
**R** <Response ID : INT8> <Num tokens : INT8> <Token data...>  
     Set tokens for datalink responses.  
     Access Restrictions: ALLOW\_DIG\_IO\_DATA\_LINK  
**S** <Digital ID : INT16>  
     Set digital ID code.  
     Access Restrictions: ALLOW\_PRODUCT\_PARAMS  
**s** <Active strobes : INT16>  
     Set active strobes mask.  
     Access Restrictions: ALLOW\_IMAGE\_PARAMS  
**W** <x1 : INT16> <y1 : INT16> <x2 : INT 16> <y2 : INT16>  
     Set window for image (will be clipped to actual image sensor dimensions during acquisition).  
     Response: actual window after clipping  
     <x1 : INT16> <y1 : INT16> <x2 : INT 16> <y2 : INT16>  
     Access Restrictions: ALLOW\_IMAGE\_PARAMS

**Pp** <Product ID : INT16>

Select power on product.  
 Access Restrictions: ALLOW\_PRODUCT\_CHANGEOVER

**PQ**

Query general product information.

Response:

<selected ID : INT16> <inspection ID : INT16>  
     <Power On Product ID : INT16>  
 <ID<sub>1</sub> : INT16> <name : string 50>  
     ...  
 <ID<sub>n</sub> : INT16> <name : string 50>

**PQ** <-1 : INT16>

Query default product image acquisition information.

Response:

<exposure time : INT32> <antiblooming : UINT4>  
     <illumination : UINT4> <delay after trigger : INT32>  
     <antiblooming level : INT16> <image gain : INT16>  
     <window x1 : INT16> <window y1 : INT16>

< window x2 : INT16> < window y2 : INT16>  
<0 : INT8> <Active Strokes : UINT16>  
<Digitizing Rate : UINT16>

**PQ** <Product ID : INT16>

Query a product for information.

Response:

<ID : INT16> <name : string 50>  
<lock count : INT8> <active : UINT4> <flash : UINT4>  
    <dirty : UINT4> <Digital ID : INT16>  
    <fail time : INT16> <OutputPulseWidth : INT16>  
    <PostOutputDelay : INT16>  
    <ResourceConflictPulseWidth : INT16>  
    <AllPassBits : UINT16>  
<exposure time : INT32> <antiblooming : UINT4>  
    <illumination : UINT4> <delay after trigger : INT32>  
    <antiblooming level : INT16> <image gain : INT16>  
    <window x1 : INT16> <window y1 : INT16>  
    < window x2 : INT16> < window y2 : INT16>  
    <0 : INT8> <Active Strokes : UINT16>  
    <Digitizing Rate : UINT16>  
<pass count : UINT32> <alarm count : UINT32>  
    <fail count : UINT32>  
<description>

**Pq** <Product ID : INT16>

Query a product for specific information.

Response:

<pass count : UINT32> <alarm count : UINT32>  
    <fail count : UINT32> <Latest Result: INT8>

**PR** <Product ID : INT16>

Reset product ID results.

**PS** <Product ID : INT16>

Select product for sensor command access by product ID.

**PSI** <Digital ID : INT16>

Select product for sensor command access by digital ID (for sensors).

**PSN** <*Name* : string 50>

Select product for sensor command access by name.

**PT** <*Product ID* : INT16> <*description* : string 200>

Set descriptive text of product.

## Register Commands

Manipulate values in register interface.

### RQ

Get the number of available registers.

**Rq** <Register : UINT16> <Mode : char>

Query value in particular register.

Response based on Mode:

| Mode | Value           |
|------|-----------------|
| B    | UINT8           |
| S    | UINT16          |
| I    | UINT32          |
| F    | FLOAT32         |
| D    | DOUBLE          |
| L    | UINT64          |
| s    | String          |
| k    | Block as UINT8s |

Note: For the block query command (k) there is an additional parameter to indicate the number of registers to query.

<Number of registers : UINT16>

**Rs** <Register : UINT16> <Mode : char>  
<Value : Mode dependent (see table)>

Set register value.

Access Restrictions: ALLOW\_REGISTER\_WRITE

## Sensor Management Commands

**SA** <*type* : INT16>

Add new sensor of *type* and auto name.

Access Restrictions: ALLOW\_SENSOR\_PARAMS

**SA** <*type* : INT16> <*name* : string 50>

Add new sensor of *type* and set name to *name*.

Response:

<*ID* : INT16> <*name* : string 50>

Access Restrictions: ALLOW\_SENSOR\_PARAMS

**SD** <*ID* : INT16>

Delete sensor.

Access Restrictions: ALLOW\_SENSOR\_PARAMS

**SI** <*ID* : INT16> <new *ID* : INT16>

Change sensor ID.

Access Restrictions: ALLOW\_SENSOR\_PARAMS

**SM**

List installed sensor modules

Response:

<*number installed modules* : INT16>

<*ID*<sub>1</sub> : INT16> <*version* : INT16> <*name* : string 50>

...

<*ID*<sub>*n*</sub> : INT16> <*version* : INT16> <*name* : string 50>

Access Restrictions: ALLOW\_SENSOR\_PARAMS

### Current image sensor modules:

| Module ID | Description        |
|-----------|--------------------|
| 1         | Edge Count         |
| 2         | Intensity          |
| 3         | Positioning        |
| 4         | Measurement        |
| 5         | Math               |
| 6         | Data Matrix Reader |
| 7         | Blob Generator     |
| 8         | Blob Selector      |
| 9         | Script             |
| 10        | Template           |
| 11        | 1-D Barcode        |
| 12        | Object Find        |
| 13        | OCR                |
| 14        | Vericode           |
| 15        | SmartLink          |
| 16        | Snowflake Reader   |
| 17        | Color Sensors      |
| 18        | Segmentation       |
| 19        | Spectrograph       |

**SN** <ID : INT16> <name : string 50>

Rename sensor.

Access Restrictions: ALLOW\_SENSOR\_PARAMS

**Sm** <Match String : string 200>

Set the match string for every reader sensor to the supplied string.

Access Restrictions: ALLOW\_SENSOR\_PARAMS

**Sm** <Sensor Name : string 200> '\t' <Match String : string 200>

Set the match string for named sensor to the supplied string.

Access Restrictions: ALLOW\_SENSOR\_PARAMS

**S0** <ID : INT16>

Query sensor output results as documented for each sensor.

**SP** <ID : INT16> <... sensor specific ...>

Set sensor parameters. See later sections on sensors.

Access Restrictions: ALLOW\_SENSOR\_SHAPE for parameter 0 (shape)  
ALLOW\_SENSOR\_PARAMS for others

**SPA** <ID : INT16> <ActiveFlag : UINT4>

Set sensor active flag.

Access Restrictions: ALLOW\_SENSOR\_PARAMS

**SPO** <ID : INT16> <Output Pass Bits : UINT16>  
<Output Warn Bits : UINT16> <Output Fail Bits : UINT16>  
<Data Link Pass Bits : UINT16> <Data Link Warn Bits : UINT16>  
<Data Link Fail Bits : UINT16>

Set output and Data Link bit pattern for that sensor.

Access Restrictions: ALLOW\_SENSOR\_PARAMS

**Sp** <Product Name : string> <'.' : char> <Sensor Name : string>  
<' ' : char> <... sensor specific ...>

Set sensor parameter for the specified sensor name within the specified product name. Set later sections on the sensors for the format of the parameters

Access Restrictions: ALLOW\_SENSOR\_SHAPE for parameter 0 (shape)  
ALLOW\_SENSOR\_PARAMS for others

**SQ**

Query general sensor information.

Response:

<ID<sub>1</sub> : INT16> <type : INT16> <name : string 50>  
...  
<ID<sub>n</sub> : INT16> <type : INT16> <name : string 50>

**SQ** <ID : INT16>

Query a sensor for information.

Response:

```
<ID : INT16> <ActiveFlag : UINT4> <name : string 50>
<pass bits : UINT16> < warn bits : UINT16>
  <fail bits : UINT16> <Data Link pass bits : UINT16>
    <Data Link warn bits : UINT16>
      <Data Link fail bits : UINT16>
<description : string 100>
< ... Processing Domain Information ... >
<Maximum Allowed Inspection Time (ms) : UINT32>
< ... Threshold Information ... >
< ... sensor specific information (see list at end) ... >
```

**Sq** <ID : INT16>

Query a sensor for information.

Response:

```
<ID : INT16> <ActiveFlag : UINT4> <name : string 50>
<LatestResult : INT8> <PassCount : UINT32> <WarnCount :
  UINT32> <FailCount : UINT32> <SpecificLastResult : INT8>
(see Sensor Input/Output Scheme for "Specific LastResult" codes)
```

**SR** <ID : INT16>

Reset sensor *ID* results.

Access Restrictions: ALLOW\_SENSOR\_PARAMS

**SR** <ID : INT16> <Product ID : INT16>

Reset sensor *ID* results (for the sensor in Product *Product ID*).

Access Restrictions: ALLOW\_SENSOR\_PARAMS

**ST** <ID : INT16> <description : string 100>

Set descriptive text of sensor

Access Restrictions: ALLOW\_SENSOR\_PARAMS

## Real-Time clock Interface

### **tNQ**

Query for status.

Response:

*<Last Error : UINT32> <Synchronized : UINT8>  
<Server Active : UINT8>*

### **tNE**

Start SNTP thread.

Access Restrictions: ALLOW\_ADMINISTRATION

### **tND**

Stop SNTP thread.

Access Restrictions: ALLOW\_ADMINISTRATION

### **tNS** <Enable Server : UINT8>

Enable or disable SNTP server.

Access Restrictions: ALLOW\_ADMINISTRATION

### **tGT**

Get time since Jan 1, 1970 00:00:00 UTC.

Response:

*<System Time : UINT32>*

### **tGZ**

Get time zone information.

Response:

*<GMT Offset : INT32> <Daylight Flag : INT8>*

GMTOffset is in minutes from GMT.

Daylight Flag:       0 DST is not in effect  
                  1 DST is in effect  
                 -1 DST is disabled

**tST** <Local Time : UINT32>

Set local time. The parameter is the number of seconds since Jan. 1, 1970.  
Access Restrictions: ALLOW\_ADMINISTRATION

**tSZ** <GMT Offset : INT32><<DST Rule Type : char>>

Set the time zone and daylight saving time information.  
The GMT Offset is in minutes from GMT.  
The DST Rule Type can be 'J' for Julian day (1-365) or 'M' for "Mm/w/d".  
DST rule is optional and supplied only if DST is used in this locale.  
The rule has the following form: *date/time date/time* dates are represented differently depending of rule type:

For 'J' type rule

- o Day: INT (1-365)

For 'M' type rule

- o Month: INT (1-12)
- o Week: INT (1-5)
- o Day of week: (0-6) (0 is Sunday)

For both rules time is represented similarly

- o Time:
  - o Hours: UINT
  - o Minutes: UINT
  - o Seconds: UINT

Note: for 'M' type rules week is not the first complete week of the month, but rather first week in which the specified day occurs. Rule M 4 1 0 is really interpreted as the "first Sunday in April". If week is 5, then the day is assumed to be the last of the month regardless of whether it falls in the third or fourth week.  
Access Restrictions: ALLOW\_ADMINISTRATION

**tSN** <STD time zone name : string 40> <DST time zone name : string 40>

Set the name for the time zones. These names are not used by firmware but are here for user interface convenience.  
Access Restrictions: ALLOW\_ADMINISTRATION

## Terminal Management Commands

**T+**

Change to selected terminal driver

**T-**

Change to terminal driver 0 (1st installed)

**Tb**

Query default terminal drivers and port information

Response:

```
<Port Number : UINT8> < Terminal ID : INT16>
<Baud Rate : INT32> <Data Bits : INT8>
<Stop Bits : INT8> <Parity : INT8> <Flow Control : INT8>
<Echo : INT8>
...
<Port Number : UINT8> < Terminal ID : INT16>
< Baud Rate : INT32> <Data Bits : INT8>
<Stop Bits : INT8> <Parity : INT8> <Flow Control : INT8>
<Echo : INT8>
```

Note:

|               |   |                       |
|---------------|---|-----------------------|
| Stop Bits:    | 0 | 1 stop bit            |
|               | 1 | 1.5 stop bits         |
|               | 2 | 2 stop bits           |
| Parity:       | 0 | No Parity             |
|               | 1 | Odd Parity            |
|               | 2 | Even Parity           |
| Flow Control: | 0 | No flow control       |
|               | 1 | XON/XOFF flow control |
|               | 2 | RTSCTS flow control   |

**Tb** <Port Number : UINT8> < Terminal ID : INT16> < Baud Rate : INT32>  
<Data Bits : INT8> <Stop Bits : INT8> <Parity : INT8>  
<Flow Control : INT8> <Echo : INT8>

Change default terminal drivers and port information

Access Restrictions: ALLOW\_COMM\_CHANGES

**TB**

Query port default terminal and baud rate.

Response:

```
<Port Number : UINT8><Terminal ID : INT16>
  <Baud Rate : INT32>
  ...
  <Port Number : UINT8> <Terminal ID : INT16>
  <Baud Rate : INT32>
```

**TB** <Port Number : UINT8> <Terminal ID : INT16> <Baud Rate : INT32>

Change default Terminal and Baud rate for *Port Number*.

**Td** <DriverID : INT8>

**TD** <DriverID : INT8>

Change serial driver( 0:TERM, 1:DVTSP)

**TL** <DriverID : INT32>

Set a link to the background terminal driver ID that was returned from the BS command. This is used to coordinate learning on a specific image.

**TmG**

Get the modbus slave enable/disable flag.

Response:

```
< Enable Flag : UINT32 >
( 1 = enabled, 0 = disabled).
```

**TmS** <Enable Flag : UINT32>

Set the modbus slave enable/disable flag. (1 = enabled, 0 = disabled).

Access Restrictions: ALLOW\_DIG\_IO\_DATA\_LINK

**TmT** <Enable Flag : INT32>

Set the modbus slave TCP keepalive timer enable/disable flag (1 = enabled, 0 = disabled). The value of the Enable flag will be saved to flash.

**TmU**

Get the value of the modbus slave TCP keepalive timer enable/disable flag.

Response:

```
<Enable Flag : INT32>
```

(1 = enabled, 0 = disabled)

**TmV** *<Enable Flag : INT32>*

Set the modbus slave TCP keepalive timer enable/disable flag. ( 1 = enabled, 0 = disabled).

Note: The value of the Enable flag will NOT be saved to flash. Note that this Command is intended to be sent by the User Interface when loading a system file In which the user has opted not to save the settings to flash. The ‘TmT’ command above is intended for use by a user since the user would presumably want to save any change in settings to flash.

**Tn** *<Destination Channel : INT8>*

**TN** *<Destination Channel : INT8>*

Create new channel using DVTSP and terminal driver 0 (1st installed).

Response:

*<New Channel Number : INT8>*

**TO**

Query Terminal timeout.

Response:

*<Terminal timeout : INT32>*

**TO** *<Terminal timeout : INT32>*

Set terminal timeout.

Access Restrictions: ALLOW\_COMM\_CHANGES

**Tq**

Return the current terminal driver.

Response:

*<Selected Terminal ID : INT16> <Terminal Name : string>*

**TQ**

Display available terminal drivers.

Response:

*<TerminalID<sub>1</sub> : INT16> <Terminal Name<sub>1</sub> : string>*

*...  
<TerminalID<sub>n</sub> : INT16> <Terminal Name<sub>n</sub> : string>*

**Tr**  
**TR**

Reset to default serial driver (TERM) and selected terminal driver.  
Access Restrictions: ALLOW\_COMM\_CHANGES

**Ts** <*DriverID* : INT16>  
**TS** <*DriverID* : INT16>

Select terminal driver.

**Tt**  
**TT**

Terminate terminal.

## System Management Commands

**Y** <+|- : char>

Enable (+) or disable (-) image acquisition/inspection process.  
Access Restrictions: ALLOW\_RUNNING\_STATE

**YD**

Query digital selection delay value.  
Response:  
    <*DSDe lay* : INT32>

**YD** <*DSDe lay* : INT32>

If *DSDe lay* < 0, Digital Selection is disabled. If *DSDe lay* >= 0, Digital Selection is enabled, and the digital ID will be read *DSDe lay* milliseconds after the Digital Select Line is activated.  
Access Restrictions: ALLOW\_DIG\_IO\_DATA\_LINK

**Ye**

Query state of Ethernet/IP driver.  
Response:  
    <Enabled : UINT32>

**Ye** <Enabled : UINT32>

Set the enable flag for the Ethernet/IP driver. This will be effective when the system is rebooted.  
Access Restrictions: ALLOW\_DIG\_IO\_DATA\_LINK

**YF**

Query if Firmware is saved in flash.  
Response:  
    <*InFlash* : UINT8>

**YI**

Trigger an inspection if the system is in external trigger mode.  
Access Restrictions: ALLOW\_RUNNING\_STATE

**YN**

Get SmartImage Sensor name.

Response:

<Name : string>

**YN** <Name : string 50>

Set SmartImage Sensor name.

Access Restrictions: ALLOW\_IP\_CONFIG

**YP** <-1|0|value : INT32>

Set image acquisition trigger source.

-1 : acquire image after old image is processed.

0 : acquire new image immediately after old image.

value : acquire image every *value* milliseconds.

Access Restrictions: ALLOW\_IMAGE\_PARAMS

**Yp** <+|- : char>

Set trigger source to internal (+) or external (-).

Access Restrictions: ALLOW\_IMAGE\_PARAMS

**YQ**

Query system status

Response:

<+|- : char> <+|- : char> <+|- : char> <period : INT32>

<cpu load : UINT8> <Unused Flash : INT32>

<System Clock : UINT32> <CCD ID : INT16 >

<InspectionProduct ID : INT16>

Running (+) / not running (-) inspections.

Internal (+) / external (-) trigger.

Digital Selection enabled (+) / disabled (-).

Trigger *period* and *CPU load* (0 - 100)% over the previous 1024 millisecond time interval.

## Yq

Query system type

Response:

<CPU Type : UINT8> <CPU Clock : UINT16>  
<FPU Present : UINT8> <Installed RAM : UINT32>  
<System Type : UINT16> <Imaging Type : UINT16>

Definitions:

|               |      |                               |
|---------------|------|-------------------------------|
| CPU Type:     | 3 :  | Motorola MPC860 or compatible |
|               | 4 :  | Intel x86                     |
|               | 5 :  | Hitachi SH-4, SH7750          |
| System Type:  | 2 :  | Model 630                     |
|               | 3 :  | Legend 520 or 530             |
|               | 4 :  | Legend 54X                    |
|               | 5 :  | SmartReader                   |
|               | 6 :  | Legend Intelligent Scanner    |
|               | 7 :  | Legend SpectroCam             |
|               |      |                               |
| Imaging Type: | 3 :  | Model 630                     |
|               | 4 :  | Legend 530 or 540             |
|               | 5 :  | Legend 520                    |
|               | 8 :  | Legend 542C                   |
|               | 11 : | Legend 544                    |
|               | 12 : | Legend 544C                   |

## YR3742

Reset SmartImage Sensor and ‘reboot’ to Diagnostics. ‘3742’ is an ASCII string.

Access Restrictions: ALLOW\_ADMINISTRATION

## YS1bdc

Save application to flash. Flash is reformatted and all flash parameters and flash heap memory is erased. ‘1bdc’ is an ASCII string.

Access Restrictions: ALLOW\_SAVE\_TO\_FLASH

## YV

Query firmware version.

Response:

<Flags : UINT8> <Major : UINT8> <Minor : UINT8>  
<Refresh : UINT8> <Build : UINT16>  
<build date : UINT32>

Flags : 0x01 Beta Release. 0x02 Alpha Release.

build date : is ddmmyyyy in hex format.

## **Debug/Test Commands**

### **Yd**

Set current channel as the debug output channel.

## Sensor Input/Output Scheme

| Type                           | Valid Values                     | Examples                                   |
|--------------------------------|----------------------------------|--------------------------------------------|
| <b>Position</b>                | x,y location and rotation vector | x,y location & rotation, equation of line. |
| <b>Pixel Value</b>             | 0-255                            | intensities                                |
| <b>Quantity</b>                | positive number                  | contrast                                   |
| <b>Percentage</b>              | 0-100                            | # edges                                    |
| <b>Boolean</b>                 | 0-1                              | # features                                 |
| <b>Statistical Numbers</b>     |                                  | # pixels                                   |
| <b>SubPixel Position</b>       |                                  | % below threshold                          |
| <b>Real Number</b>             |                                  | go/no-go, presence                         |
| <b>String</b>                  |                                  | equation of line or circle                 |
| <b>Vectors</b>                 |                                  | angle, roundness                           |
| <b>Integer Lists</b>           |                                  | Bar code string                            |
| <b>SubPixel Position Lists</b> |                                  | Blob feature vector                        |
| <b>Real Number Lists</b>       |                                  | Blob areas                                 |
|                                |                                  | Blob positions                             |
|                                |                                  | Blob angles                                |

### Specific Input/Output Numbers

| Output                   | Decimal Number | Structure Values |         |
|--------------------------|----------------|------------------|---------|
|                          |                | ID               | Param   |
| <b>Position</b>          | 0              | 0                | X       |
|                          |                | 1                | Y       |
|                          |                | 2                | Theta X |
|                          |                | 3                | Theta Y |
| <b>Minimum Intensity</b> | 16             |                  |         |
| <b>Maximum Intensity</b> | 17             |                  |         |
| <b>Mean Intensity</b>    | 18             |                  |         |
| <b>Median Intensity</b>  | 19             |                  |         |
| <b>Contrast</b>          | 20             |                  |         |
| <b>Threshold</b>         | 21             |                  |         |
| <b># of Edges</b>        | 32             |                  |         |
| <b># of Features</b>     | 33             |                  |         |
| <b># of Pixels</b>       | 34             |                  |         |
| <b># of Blobs</b>        | 35             |                  |         |
| <b>% below threshold</b> | 48             |                  |         |
| <b>Template % error?</b> | 49             |                  |         |
| <b>SubPixel Point</b>    | 96             | 0                | X       |
|                          |                | 1                | Y       |

|                                    |     |     |        |
|------------------------------------|-----|-----|--------|
| <b>SubPixel Line</b>               | 97  | 0   | A      |
|                                    |     | 1   | B      |
|                                    |     | 2   | Xo     |
|                                    |     | 3   | Yo     |
| <b>SubPixel Circle</b>             | 98  | 0   | X      |
|                                    |     | 1   | Y      |
|                                    |     | 2   | Radius |
| <b>Distance/Radius</b>             | 112 |     |        |
| <b>Angle</b>                       | 113 |     |        |
| <b>Straightness</b>                | 114 |     |        |
| <b>Roundness</b>                   | 115 |     |        |
| <b>Scale Factor</b>                | 116 |     |        |
| <b>Bar Code String</b>             | 128 |     |        |
| <b>Blob Feature Vector</b>         | 144 |     |        |
| <b>Blob 1 Area</b>                 | 256 |     |        |
| ...                                | ... | ... |        |
| <b>Blob 8 Area</b>                 | 263 |     |        |
| <b>Blob 1 Subpix Pos</b>           | 272 |     |        |
| ...                                | ... | ... |        |
| <b>Blob 8 Subpix Pos</b>           | 279 |     |        |
| <b>Blob 1 Angle</b>                | 288 |     |        |
| ...                                | ... | ... |        |
| <b>Blob 8 Angle</b>                | 295 |     |        |
| <b>Coordinate System</b>           | 304 |     |        |
| <b>Precision</b>                   | 372 |     |        |
| <b>Blob Area (INT32)</b>           | 512 |     |        |
| <b>Blob Subpix Pos (FPOINT)</b>    | 513 |     |        |
| <b>Blob Angle (FLOAT64)</b>        | 514 |     |        |
| <b>Average Intensity (INT32)</b>   | 515 |     |        |
| <b>Bounding Box Width (INT32)</b>  | 516 |     |        |
| <b>Bounding Box Height (INT32)</b> | 517 |     |        |
| <b>Bounding Box Window (RECT)</b>  | 518 |     |        |
| <b>Perimeter (FLOAT64)</b>         | 519 |     |        |
| <b>Compactness (FLOAT64)</b>       | 520 |     |        |
| <b>Eccentricity (FLOAT64)</b>      | 521 |     |        |
| <b>Max Point (POINT)</b>           | 522 |     |        |
| <b>M10 (FLOAT64)</b>               | 523 |     |        |

|                                                           |      |  |
|-----------------------------------------------------------|------|--|
| <b>M01 (FLOAT64)</b>                                      | 524  |  |
| <b>M20 (FLOAT64)</b>                                      | 525  |  |
| <b>M11 (FLOAT64)</b>                                      | 526  |  |
| <b>M02 (FLOAT64)</b>                                      | 527  |  |
| <b>M30 (FLOAT64)</b>                                      | 528  |  |
| <b>M21 (FLOAT64)</b>                                      | 529  |  |
| <b>M12 (FLOAT64)</b>                                      | 530  |  |
| <b>M03 (FLOAT64)</b>                                      | 531  |  |
| <b>PA Angle (FLOAT64)</b>                                 | 532  |  |
| <b>PawS Angle<br/>(FLOAT64)</b>                           | 533  |  |
| <b>MaxPoint Angle<br/>(FLOAT64)</b>                       | 534  |  |
| <b>Radius (FLOAT64)</b>                                   | 535  |  |
| <b>Transformed Coord<br/>(FPOINT)</b>                     | 1096 |  |
| <b>Transformed Coord<br/>(circle center)<br/>(FPOINT)</b> | 1098 |  |
| <b>Transformed Coord<br/>(blob centroid)<br/>(FPOINT)</b> | 1513 |  |

### Specific SoftSensor Result Codes

- Negative numbers are failure codes.
- Positive numbers are warning codes.
- Zero indicates the sensor passed the inspection.

Codes for pass/warn violations are from abs(1 to 63). General failure codes are from abs(64 and up)

| Failure Code | All Sensors                  |
|--------------|------------------------------|
| -64          | Unable to allocate memory    |
| -65          | SoftSensor out of bounds     |
| -66          | Bad position reference       |
| -67          | Bad low intensity reference  |
| -68          | Bad High intensity reference |
| -69          | Bad Feature vector reference |

| Failure Code | Warning Code | EdgeCount SoftSensors                                |
|--------------|--------------|------------------------------------------------------|
| -1           | 1            | min contrast not met ((failing) value < min # edges) |
| -2           | 2            | value < minimum number of edges                      |
| -3           | 3            | value > maximum number of edges                      |
| -4           | 4            | value < maximum passing percent (scanning sensor)    |

| Failure Code | Warning Code | FeatureCount SoftSensors                                |
|--------------|--------------|---------------------------------------------------------|
| -1           | 1            | min contrast not met ((failing) value < min # features) |
| -2           | 2            | value < minimum number of features                      |
| -3           | 3            | value > maximum number of features                      |
| -4           | 4            | value < maximum passing percent (scanning sensor)       |

| Failure Code | Warning Code | Intensity SoftSensors                             |
|--------------|--------------|---------------------------------------------------|
| -1           | 1            | value < minimum contrast                          |
| -2           | 2            | value > maximum contrast                          |
| -3           | 3            | value < minimum percent of pixels below threshold |
| -4           | 4            | value > maximum percent of pixels below threshold |

| Failure Code | Warning Code | Positioning (Translational/Rotational) SoftSensors |
|--------------|--------------|----------------------------------------------------|
| -1           | 1            | value < minimum contrast                           |
| -2           | 2            | value < minimum percent offset1                    |
| -3           | 3            | value > maximum percent offset1                    |
| -4           | 4            | value < minimum percent offset2                    |
| -5           | 5            | value > maximum percent offset2                    |
| -6           | 6            | bad line fit (area edge)                           |
| -7           |              | no intersection (area edge)                        |
| -8           |              | no edge found (lines, arcs)                        |

| Failure Code | Warning Code | Measurement SoftSensors      |
|--------------|--------------|------------------------------|
| -1           | 1            | value < minimum contrast     |
| -2           | 2            | value < minimum distance     |
| -3           | 3            | value > maximum distance     |
| -4           | 4            | value < minimum radius/angle |
| -5           | 5            | value > maximum radius/angle |
| -6           | 6            | not enough edges found       |
| -7           |              | no intersection (area edge)  |
| -8           |              | straightness/roundness       |

| Failure Code | Warning Code | Math SoftSensors       |
|--------------|--------------|------------------------|
| -1           | 1            | minimum distance/angle |
| -2           | 2            | maximum distance/angle |

| Failure Code | Warning Code | DataMatrix SoftSensors              |
|--------------|--------------|-------------------------------------|
| -1           | 1            | String does not match pass criteria |

| Failure Code | Warning Code | Blob Generation SoftSensors |
|--------------|--------------|-----------------------------|
| -1           | 1            | Minimum contrast            |
| -2           | 2            | Minimum blobs               |
| -3           | 3            | Maximum blobs               |

| Failure Code | Warning Code | Blob Selection SoftSensors |
|--------------|--------------|----------------------------|
| -1           | 1            | Minimum blobs              |
| -2           | 2            | Maximum blobs              |

| Failure Code | Warning Code | Scripting SoftSensors                    |
|--------------|--------------|------------------------------------------|
| -1           | 1            | Result set to FAIL or WARN               |
| -2           | 2            | Processing error                         |
| <-2          | >2           | Result set to unknown fail or warn value |

## Setting Thresholds for Any Sensor

### Command Format For Setting Thresholds

**SP** <sensor ID : INT16> <parameter ID: UINT8> <Threshold Type: INT8>  
<data>

NOTE: The data field includes the new threshold level scaled by 100. So if the user wants to set the threshold level at 80, then the data field should include an 8000 (80 scaled up by 100). Likewise, when a min contrast must be specified, it has to be scaled up by 100.

<parameter ID> = 251

<Threshold Type>

- 1 Invalid threshold, no data to follow
- 0 Fixed threshold, followed by threshold Value of UINT16
- 1 % Contrast threshold, followed by threshold Value of UINT16

Example: set percent contrast threshold for a sensor with ID = 1 at a level of 45

#SP 1 251 1 4500

- 32 Fixed Gradient, followed by threshold Value of UINT16
- 33 % Contrast Gradient, followed by threshold Value of UINT16
- 2 % Reference, followed by:

UINT16 threshold Value  
INT16 Low Int Sensor ID  
INT16 Low Int Value ID  
INT16 High Int Sensor ID  
INT16 High Int Value ID

Example: set percent of reference intensities threshold for a sensor with ID = 1 reading the maximum (code 17) and the minimum (code 16) intensities from a sensor with ID = 5 at a level of 80%

#SP 1 251 2 8000 5 16 5 17

- 34 % Reference Gradient, followed by:  
UINT16 threshold Value  
INT16 Low Int Sensor ID  
INT16 Low Int Value ID  
INT16 High Int Sensor ID  
INT16 High Int Value ID
- 3 Auto Bimodal, followed by Ignore Saturation of UINT8
- 4 Adaptive, followed by  
UINT16 Value  
INT16 Window Size  
UINT16 Min Contrast
- 6 Dominant Color, followed by all color data
- 64 OCR computed, followed by:  
UINT16 Value  
UINT8 Use Float  
UINT8 Extended  
UINT8 BlockSize

|    |                              |
|----|------------------------------|
| 65 | OCR Linear, followed by:     |
|    | UINT16      Value            |
|    | UINT8        Use Float       |
|    | UINT8        Extended        |
|    | UINT8        BlockSize       |
| 66 | OCR Non-Linear, followed by: |
|    | UINT16      Value            |
|    | UINT8        Use Float       |
|    | UINT8        Extended        |
|    | UINT8        BlockSize       |

## Query Response of Thresholds for any Sensor

<Threshold Type : INT8> followed by <data>

<Threshold Type>

|    |                                                             |
|----|-------------------------------------------------------------|
| -1 | Invalid threshold, no data to follow                        |
| 0  | Fixed threshold, followed by threshold Value of UINT16      |
| 1  | % Contrast threshold, followed by threshold Value of UINT16 |
| 32 | Fixed Gradient, followed by threshold Value of UINT16       |
| 33 | % Contrast Gradient, followed by threshold Value of UINT16  |
| 2  | % Reference, followed by:                                   |
|    | UINT16      threshold Value                                 |
|    | INT16        Low Int Sensor ID                              |
|    | INT16        Low Int Value ID                               |
|    | INT16        High Int Sensor ID                             |
|    | INT16        High Int Value ID                              |
| 34 | % Reference Gradient, followed by:                          |
|    | UINT16      threshold Value                                 |
|    | INT16        Low Int Sensor ID                              |
|    | INT16        Low Int Value ID                               |
|    | INT16        High Int Sensor ID                             |
|    | INT16        High Int Value ID                              |
| 3  | Auto Bimodal, followed by Ignore Saturation of UINT8        |
| 4  | Adaptive, followed by                                       |
|    | UINT16      Value                                           |
|    | INT16        Window Size                                    |
|    | UINT16      Min Contrast                                    |
| 6  | Dominant Color, followed by all color data                  |
| 64 | OCR computed, followed by:                                  |
|    | UINT16      Value                                           |
|    | UINT8        Use Float                                      |
|    | UINT8        Extended                                       |
|    | UINT8        BlockSize                                      |
| 65 | OCR Linear, followed by:                                    |
|    | UINT16      Value                                           |
|    | UINT8        Use Float                                      |
|    | UINT8        Extended                                       |

|    |                              |           |
|----|------------------------------|-----------|
|    | UINT8                        | BlockSize |
| 66 | OCR Non-Linear, followed by: |           |
|    | UINT16                       | Value     |
|    | UINT8                        | Use Float |
|    | UINT8                        | Extended  |
|    | UINT8                        | BlockSize |

## Setting Common Parameters for any sensor

### **Command Format For Displaying Active Plane on Color cameras**

**SP** <sensor ID : INT16> <(parameter ID) 252 : UINT8>

### **Command Format For Setting Max Inspection Time**

**SP** <sensor ID : INT16> <(parameter ID) 253: UINT8>  
 <Maximum Time : UINT32>

### **Command Format For Changing Dominant Color threshold**

**SP** <sensor ID : INT16> <(parameter ID) 254 : UINT8>  
 <Color Sensor ID : INT16> <Color Number : INT32>

This command copies the specified color number from specified color sensor ID into Dominant color of sensor with sensor ID.

### **Command Format For Setting Processing Domain**

**SP** <sensor ID : INT16> <(parameter ID) 255 : UINT8>  
 <Active Plane : INT32> <Red Coef : INT32> <Green Coef : INT32>  
 <Blue Coef : INT32>

|                  |    |        |
|------------------|----|--------|
| <Active Plane> = | -1 | Color  |
|                  | 0  | Custom |
|                  | 1  | Red    |
|                  | 2  | Green  |
|                  | 3  | Blue   |
|                  | 4  | System |

### **Command Format For Setting Learned Position**

**SP** <sensor ID : INT16> <(parameter ID) 199 : UINT8>  
 <Learned Position X : INT32> <Learned Position Y : INT32>  
 <Learned Position Theta 0 : INT32>  
 <Learned Position Theta 1 : INT32>

### **Command Format For Setting Position Reference**

**SP** <sensor ID : INT16> <(parameter ID) 200 : UINT8>  
 <Reference Sensor ID : INT16> <Reference Output ID : INT16>



|    |                                                                 |                                                                                                                                                                                      |
|----|-----------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 19 | Double rectangle                                                | <Rectangle1 : Rectangle><br><Rectangle2 : Rectangle>                                                                                                                                 |
| 20 | Annulus Arc                                                     | <Center : Point><br><Radius : UINT32><br><Depth : INT32><br><Arc point1 : Point>      start angle<br><Arc point2 : Point>      end angle<br><Direction : INT8>      -1 : CCW, 1 : CW |
| 20 | Measurement Area<br>(Parallelogram scanning in both directions) | See parallelogram                                                                                                                                                                    |
| 21 | Parametric Arc                                                  | <Arc : Elliptical Arc>                                                                                                                                                               |
| 22 | Parametric Circular arc                                         | <Arc : Elliptical Arc> Circular arc assumes Radius2 is equal to Radius1. If it is not, the parameter is ignored.                                                                     |
| 10 | Filled Rectangle                                                | see Rectangle                                                                                                                                                                        |
| 11 | Filled Parallelogram                                            | see Parallelogram                                                                                                                                                                    |
| 12 | Filled Ellipse                                                  | see Ellipse                                                                                                                                                                          |
| 13 | Filled Polygon                                                  | see Polygon                                                                                                                                                                          |

Note: Filled Rectangle, Parallelogram, Ellipse, and Polygon take the same arguments as their contour variants. The only difference is that sensor treats ‘filled’ types as area shapes and ‘unfilled’ as contour shapes.

Example: Reshape an elliptical area sensor with ID 1 to have the center at (250, 300) semi-major axis of 75 and semi-minor axis of 50.

```
#SP 1 0 12 250 300 325 300 250 350
```

The command includes the sensor ID (1) the parameter ID for shape change (0) and the shape data (for ellipse, center point and two more points on the ellipse defining the radii).

### ***Specific shapes associated with every SoftSensor***

The tables below indicate the type of shape that is associated with every SoftSensor. The numbers in the second column refer to the parameter ID as shown above. Use both tables in order to configure the commands correctly.

| SoftSensor Type        | SoftSensor Shape       | Shape ID |
|------------------------|------------------------|----------|
| Translation            | Translation Line       | 1        |
|                        | Fiducial Square        | 3        |
|                        | Fiducial Rectangle     | 3        |
|                        | Fiducial Ellipse       | 5        |
|                        | Fiducial Circle        | 5        |
| Rotation               | Circular Arc           | 2        |
|                        | Elliptical Arc         | 2        |
|                        | Parallelogram          | 4        |
| Intensity              | Lines                  | 9        |
|                        | Polyline               | 9        |
|                        | Freehand Line          | 9        |
|                        | Square                 | 9        |
|                        | Rectangle              | 9        |
|                        | Polygon                | 9        |
|                        | Circle                 | 2        |
|                        | Ellipse                | 2        |
|                        | Circule Arc            | 2        |
|                        | Elliptical Arc         | 2        |
|                        | Square Area            | 13       |
|                        | Rectangular Area       | 13       |
|                        | Polygon Area           | 13       |
|                        | Circular Area          | 12       |
|                        | Elliptical Area        | 12       |
| EdgeCount/FeatureCount | Lines                  | 1        |
|                        | Polyline               | 1        |
|                        | Freehand Line          | 1        |
|                        | Square                 | 1        |
|                        | Rectangle              | 1        |
|                        | Polygon                | 1        |
|                        | Circle                 | 2        |
|                        | Ellipse                | 2        |
|                        | Circular Arc           | 2        |
|                        | Elliptical Arc         | 2        |
| Measurement            | Lines                  | 1        |
|                        | Measure Across Line    | 17       |
|                        | FindEdge Parallelogram | 4        |
|                        | Measure Across Area    | 20       |
|                        | FindCircle             | 18       |
|                        | Segmented FindCircle   | 18       |
| Blob Generator         | Rectangle              | 10       |
|                        | Ellipse                | 12       |
|                        | Polygon                | 13       |

|                                                                                                                                                                                      |                   |    |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|----|
| Template Match                                                                                                                                                                       | Rectangle*        | 0  |
|                                                                                                                                                                                      | Double Rectangle* | 1  |
| * Requires two points for each rectangle even though the description of the shape is not defined as a rectangle (actual codes 3 and 19 for single and double rectangle respectively) |                   |    |
| ObjectFind                                                                                                                                                                           | Rectangle         | 3  |
|                                                                                                                                                                                      | Double Rectangle  | 19 |
|                                                                                                                                                                                      | Polygon           | 9  |
|                                                                                                                                                                                      | Ellipse           | 5  |
| Pixel Counting                                                                                                                                                                       | Square            | 10 |
|                                                                                                                                                                                      | Rectangle         | 10 |
|                                                                                                                                                                                      | Circle            | 12 |
|                                                                                                                                                                                      | Ellipse           | 12 |
|                                                                                                                                                                                      | Polygon           | 13 |
| Color Monitoring                                                                                                                                                                     | Square            | 10 |
|                                                                                                                                                                                      | Rectangle         | 10 |
|                                                                                                                                                                                      | Circle            | 12 |
|                                                                                                                                                                                      | Ellipse           | 12 |
|                                                                                                                                                                                      | Polygon           | 13 |
| Segmentation                                                                                                                                                                         | Lines             | 9  |
|                                                                                                                                                                                      | Polyline          | 9  |
|                                                                                                                                                                                      | Freehand Line     | 9  |
|                                                                                                                                                                                      | Square            | 9  |
|                                                                                                                                                                                      | Rectangle         | 9  |
|                                                                                                                                                                                      | Polygon           | 9  |
|                                                                                                                                                                                      | Circle            | 2  |
|                                                                                                                                                                                      | Ellipse           | 2  |
|                                                                                                                                                                                      | Circular Arc      | 2  |
|                                                                                                                                                                                      | Elliptical Arc    | 2  |
|                                                                                                                                                                                      | Square Area       | 10 |
|                                                                                                                                                                                      | Rectangle Area    | 10 |
|                                                                                                                                                                                      | Circle Area       | 12 |
|                                                                                                                                                                                      | Ellipse Area      | 12 |
|                                                                                                                                                                                      | Polygon Area      | 13 |

|         |                          |    |
|---------|--------------------------|----|
| Readers | DataMatrix Rectangle     | 3  |
|         | DataMatrix Parallelogram | 4  |
|         | DataMatrix Ellipse       | 5  |
|         | DataMatrix Polygon       | 9  |
|         | BarCode Line             | 1  |
|         | BarCode Arc              | 22 |
|         | BarCode Parallelogram    | 4  |
|         | OCR Rectangle            | 3  |
|         | OCR Parallelogram        | 4  |
|         | OCR Arc                  | 20 |
|         | SnowFlake Rectangle      | 3  |
|         | SnowFlake Ellipse        | 5  |

## Edge Counting Sensors (Module ID: 1)

### Command Format For Setting EdgeCounting Sensor Parameters

**SP** <sensor ID : INT16> <parameter ID : UINT8> <data>

<parameter ID>:

- 0 Set Sensor Shape
- 2 Set feature parameters  
    <data>  $\Rightarrow$  <edge type : UINT4> <min light size :  
    UINT16> <max light size : UINT16> <min dark size :  
    UINT16> <max dark size : UINT16> <Require Alternating  
    Features : UINT8> <Enable End Features : UINT8>  
    <<...>> are for feature counting.
- 3 Set scanning parameter  
    <data>  $\Rightarrow$  <x vector component : INT16> <y vector  
    component: INT16> <main vector number of steps:  
    INT16> <main vector scan density : INT16>  
    <perpendicular vector number of steps : INT16>  
    <perpendicular scan density : INT16> <output type :  
    INT 16>

Comments:

- vector components are for the main direction vector and are entered as a unit vector scaled by 16384 ( $2^{14}$ )
- the number of steps are in the positive x direction for the main vector and in the y direction for the perpendicular vector. Then the coordinates are rotated to the vector direction entered above. However, the number of steps are taken in both the positive and negative directions.
- The scan density is similar to the scan density for area edge scanning and is entered as a number from 0 to 10000.
- Output Type:
  - 0 = check passing percent
  - 1 = check minimum number of edges versus the constraints
  - 2 = check maximum number of edges versus the constraints

- 8 Set sensor warning levels  
    <data>  $\Rightarrow$  <min contrast : INT16> <min edges : INT16>  
    <max edges : INT16> < min percent passing : INT16>
- 9 Set sensor passing levels  
    <data>  $\Rightarrow$  <min contrast : INT16> <min edges : INT16>  
    <max edges : INT16> < min percent passing : INT16>

## EdgeCounting Sensor Query Response

<type/shape : UINT8> (bit 7 set if feature counting)

0 No shape defined

1 PolyLine

<num points>

<x<sub>1</sub>><y<sub>1</sub>> ... <x<sub>10</sub>><y<sub>10</sub>>

...

<x<sub>n-9</sub>><y<sub>n-9</sub>> ... <x<sub>n</sub>><y<sub>n</sub>>

2 Ellipse

<cx> <cy> <x<sub>1</sub>> <y<sub>1</sub>> <x2> <y2> <<xa1><ya1><xa2><ya2>>

<threshold type : UINT4> <value : UINT8>

0 Fixed greyscale value

1 Percent of contrast along path

2 Percent of reference input contrast

<edge type : UINT4> <<min light size : UINT16> <max light size :  
UINT16> <min dark size : UINT16> <max dark size : UINT16>  
<Require Alternating Features : UINT8> <Force End  
Transitions : UINT8>>

edge type:

0 Either

1 Positive edge / bright fiducial

2 Negative edge / dark fiducial

scanning parameter: <x vector component : INT16> <y vector component :  
INT16> <main vector number of steps: INT16> <main vector scan  
density : INT16> <perpendicular vector number of steps : INT16>  
<perpendicular scan density : INT16> <output type : INT16>

output type:

0 Percent passing

1 Minimum number of edges

2 Maximum number of edges

warning levels: <min contrast : INT16> <min edges : INT16> <max edges :  
INT16> <min percent passing while scanning : INT16>

pass levels: <min contrast : INT16> <min edges : INT16> <max edges :  
INT16> <min percent passing while scanning : INT16>

Input References: <position - sensor ID:INT16> <position - output  
ID:INT16> <low intensity - sensor ID:INT16> <output ID:INT16> <  
high intensity - sensor ID: INT16> <output ID : INT16>

Position during Learning: <x : INT32><y: INT32><theta : FIXED32, FIXED32>

| Edge/Feature Count Sensor Referencing IDs       |    |
|-------------------------------------------------|----|
| Input (i.e. – Will reference other sensors by): |    |
| Position                                        | 0  |
| Low Intensity Reference                         | 16 |
| High Intensity Reference                        | 17 |
| Output (i.e. – Can be referenced by)            |    |
| Min Intensity                                   | 16 |
| Max Intensity                                   | 17 |
| Contrast                                        | 20 |
| Threshold                                       | 21 |
| Number Edges                                    | 32 |

| Failure Code | Warning Code | EdgeCount SoftSensors                                |
|--------------|--------------|------------------------------------------------------|
| -1           | 1            | min contrast not met ((failing) value < min # edges) |
| -2           | 2            | value < minimum number of edges                      |
| -3           | 3            | value > maximum number of edges                      |
| -4           | 4            | value < maximum passing percent (scanning sensor)    |

| Failure Code | Warning Code | FeatureCount SoftSensors                                |
|--------------|--------------|---------------------------------------------------------|
| -1           | 1            | min contrast not met ((failing) value < min # features) |
| -2           | 2            | value < minimum number of features                      |
| -3           | 3            | value > maximum number of features                      |
| -4           | 4            | value < maximum passing percent (scanning sensor)       |

### EdgeCounting Sensor Output Query Response

Outputs: <min intensity : INT32> <max intensity : INT32> <contrast : INT32> <threshold : INT32> <num edges : INT32> <Percent Passing : INT32>

For scanning, *num edges* corresponds to the min or maximum number of edges if the output type is min or max edges respectively. *num edges* is -1 if output type is percent passing.

For non-scanning, *Percent Passing* is -1.

## Intensity Sensors (Module ID: 2)

### Command Format For Setting Intensity Reference Sensor Parameters

**SP** <sensor ID : INT16><parameter ID : UINT8><data>

<parameter ID>:

0 Set sensor type and coordinates

<data> ⇒ <type : UINT8> <coord data>

<type>:

1 Poly-line

<coord data> ⇒ <num points : INT16> <x1 : INT16> <y1 : INT16> ... <x<sub>n</sub> : INT16> <y<sub>n</sub> : INT16>

2 Elliptical arc

<coord data> ⇒ <xc : INT16> <yc : INT16>  
<xr1 : INT16> <yr1 : INT16> <xr2 : INT16>  
<yr2 : INT16> <<xa1 : INT16> <ya1 : INT16>  
<xa2 : INT16> <ya2 : INT16>>

<<...>> are optional - for open arc.

3 Rectangle (area)

<coord data> ⇒ <x1 : INT16> <y1 : INT16> <x2 : INT16>  
<y2 : INT16>

5 Ellipse (area)

<coord data> ⇒ <xc : INT16> <yc : INT16> <xr1 : INT16> <yr1 : INT16> <xr2 : INT16> <yr2 : INT16>

9 PolyLine

<coord data> ⇒ <num points : INT16> <x1 : INT16> <y1 : INT16> ... <x<sub>n</sub> : INT16> <y<sub>n</sub> : INT16>

2 Set sampling parameters

<data> ⇒ <x sampling density : INT16 (0-10000)> <y sampling density : INT16 (0-10000)>

8 Set sensor warning levels

<data> ⇒ <min contrast : INT16> <max contrast : INT16> <min % pixel below threshold : INT16> <max % pixel below threshold : INT16>

9 Set sensor passing levels

<data> ⇒ <min contrast : INT16> <max contrast : INT16> <min % pixel below threshold : INT16> <max % pixel below threshold : INT16>

## **Intensity Reference Sensor Query Response**

*<type/shape : UINT8>* (bit 7 high for web check sensor)

- 0 No shape defined
- 1 PolyLine
  - <number points>*
  - <x<sub>1</sub>><y<sub>1</sub>> ... <x<sub>10</sub>><y<sub>10</sub>>*
  - ...*
  - <x<sub>n-9</sub>><y<sub>n-9</sub>> ... <x<sub>n</sub>><y<sub>n</sub>>*
- 2 Elliptical arc
  - <xc><yc><xr1> <yr1><xr2><yr2><<xa1><ya1><xa2><ya2>>*
- 3 Rectangle
  - <x1> <y1> <x2> <y2>*
- 5 ellipse
  - <xc> <yc> <xr1> <yr1> <xr2> <yr2>*

*<threshold type : UINT4> <value : UINT8>*

- 0 fixed greyscale value
- 1 percent of contrast along path
- 2 percent of reference input contrast

*<x\_sampling density : INT16 (0-10000)> <y sampling density : INT16 (0-10000)>*

warning levels: *<min contrast : INT16> <max contrast : INT16> <min %pixels below threshold : INT16> <max %pixel below threshold : INT16>*

pass levels: *<min contrast : INT16> <max contrast : INT16> <min %pixel below threshold : INT16> <max %pixel below threshold : INT16>*

Input References: *<position - sensor ID:INT16> <position - output ID:INT16>*  
*<LowIntensity Reference - sensor ID:INT16>.<LowIntensity Reference - output ID:INT16>.<HighIntensity Reference - sensor output ID:INT16> <HighIntensity Reference - output ID:INT16>*

Outputs: *<min intensity : INT32><max intensity : INT32>*  
*<MeanIntensity : INT32> <MedianIntensity : INT32>*  
*<contrast : INT32> <threshold : INT32> <pixel count : INT32>*

Position during Learning: *<x : INT32><y : INT32><theta : FIXED32, FIXED32>*

| <b>Intensity Sensor Referencing IDs</b>         |    |
|-------------------------------------------------|----|
| Input (i.e. – Will reference other sensors by): |    |
| Position                                        | 0  |
| Low Intensity Reference                         | 16 |
| High Intensity Reference                        | 17 |
| Output (i.e. – Can be referenced by)            |    |
| Min Intensity                                   | 16 |
| Max Intensity                                   | 17 |
| Contrast                                        | 20 |
| Threshold                                       | 21 |
| % Below Threshold                               | 48 |

| <b>Failure Code</b> | <b>Warning Code</b> | <b>Intensity SoftSensors</b>                      |
|---------------------|---------------------|---------------------------------------------------|
| -1                  | 1                   | value < minimum contrast                          |
| -2                  | 2                   | value > maximum contrast                          |
| -3                  | 3                   | value < minimum percent of pixels below threshold |
| -4                  | 4                   | value > maximum percent of pixels below threshold |

## Positioning Sensors (Module ID: 3)

### Command Format For Setting Positioning Sensor Parameters

{...} are for area based positioning sensors. [...] are for area-edge positioning sensor.

**SP** <sensor ID : INT16><parameter ID : UINT8><data>

<parameter ID> :

- 0 Set sensor Shape  
<data>  $\Rightarrow$  <type : UINT8><coord data>  
<type>
  - 1 straight line  
<coord data>  $\Rightarrow$  <x1 : INT16> <y1 : INT16> <x2 : INT16> <y2 : INT16>
  - 2 elliptical arc  
<coord data>  $\Rightarrow$  <xc : INT16> <yc : INT16> <xr1 : INT16>  
<yr1 : INT16> <xr2 : INT16> <yr2 : INT16> <<xa1 : INT16>  
<ya1 : INT16> <xa2 : INT16> <ya2 : INT16>>  
<<...>> are optional - for open arc.
  - 3 rectangle (fiducial)  
<coord data>  $\Rightarrow$  <x1 : INT16> <y1 : INT16> <x2 : INT16> <y2 : INT16>
  - 4 area-edge sensor  
<coord data>  $\Rightarrow$  <x1 : INT16> <y1 : INT16> <x2 : INT16> <y2 : INT16> <x3 : INT16> <y3 : INT16>
  - 5 elliptical area  
<coord data>  $\Rightarrow$  <xc : INT16> <yc : INT16> <xr1 : INT16>  
<yr1 : INT16> <xr2 : INT16> <yr2 : INT16>
- 2 set feature parameters  
<data>  $\Rightarrow$  <edge/feature type : UINT4> <feature parameters>  
  
edge type:
  - 0 either
  - 1 positive edge / bright fiducial
  - 2 negative edge / dark fiducialscan sides [bit 3, 2, 1, 0]  $\Rightarrow$  [right, bottom, left, top]

<feature parameters>  $\Rightarrow$

straight line

<edge index : UINT16> <Minimum Light Feature Size :  
UINT16> <Maximum Light Feature Size : UINT16>  
<Minimum Dark Feature Size : UINT16> <Maximum Dark  
Feature Size : UINT16> <Require Alternating Feature :  
UINT4> <Enable End Feature : UINT4>

elliptical arc

<edge index : UINT16> <Minimum Light Feature Size :  
UINT16> <Maximum Light Feature Size : UINT16>

*<Minimum Dark Feature Size : UINT16> <Maximum Dark Feature Size : UINT16> <Require Alternating Feature : UINT4> <Enable End Feature : UINT4>*

rectangle (fiducial)

*<scan sides : UINT4>*

area-edge sensor

*<edge index : UINT16> <Minimum Light Feature Size : UINT16> <Maximum Light Feature Size : UINT16> <Minimum Dark Feature Size : UINT16> <Maximum Dark Feature Size : UINT16> <Require Alternating Feature : UINT4> <Enable End Feature : UINT4> <scan density : INT16>*

elliptical area

*<scan sides : UINT4>*

8 set sensor warning levels

*<data> ⇒ <min contrast : INT16> <min offset : INT16> <max offset : INT16> {<min y offset 2 : INT16> <max y offset : INT16>} [<min theta : INT16> <max theta : INT16>]*

9 set sensor passing levels

*<data> ⇒ <min contrast : INT16> <min offset : INT16> <max offset : INT16> {<min y offset 2 : INT16> <max y offset : INT16>} [<min theta : INT16> <max theta : INT16>]*

## Positioning Sensor Query Response

<type/shape : UINT8>

- 0 no shape defined
- 1 straight line  
    <x1> <y1> <x2> <y2>
- 2 elliptical arc  
    <cx> <cy> <x1> <y1> <x2> <y2> <<xa1><ya1><xa2><ya2>>
- 3 rectangular area  
    <x1> <y1> <x2> <y2>
- 4 area-edge locator  
    <x1> <y1> <x2> <y2> <x3> <y3> <x4> <y4>
- 5 elliptical area  
    <cx> <cy> <x1> <y1> <x2> <y2>

<threshold type : UINT4> <value : UINT8>

- 0 fixed grayscale value
- 1 percent of contrast along path
- 2 percent of reference input contrast

<edge type : UINT4>...

straight line

    <edge index : UINT16> <Minimum Light Feature Size : UINT16>  
    <Maximum Light Feature Size : UINT16> <Minimum Dark Feature  
    Size : UINT16> <Maximum Dark Feature Size : UINT16> <Require  
    Alternating Feature : UINT4> <Enable End Feature : UINT4>

elliptical arc

    <edge index : UINT16> <Minimum Light Feature Size : UINT16>  
    <Maximum Light Feature Size : UINT16> <Minimum Dark Feature  
    Size : UINT16> <Maximum Dark Feature Size : UINT16> <Require  
    Alternating Feature : UINT4> <Enable End Feature : UINT4>

rectangle (fiducial)

    <scan sides : UINT4>

area-edge sensor

    <edge index : UINT16> <Minimum Light Feature Size : UINT16>  
    <Maximum Light Feature Size : UINT16> <Minimum Dark Feature  
    Size : UINT16> <Maximum Dark Feature Size : UINT16> <Require  
    Alternating Feature : UINT4> <Enable End Feature : UINT4>  
    <scan density : INT16>

elliptical area

    <scan sides : UINT4>

edge type:

- 0 either
  - 1 positive edge / bright fiducial
  - 2 negative edge / dark fiducial
- scan sides [bit 3, 2, 1, 0] ⇒ [right, bottom, left, top]

warning levels:

```
<min contrast : INT16> <min offset : INT16> <max offset :  
INT16> {<min y offset : INT16> <max y offset : INT16>}  
[<min theta : INT16> <max theta : INT16>]
```

failure levels:

```
<min contrast : INT16> <min offset : INT16> <max offset :  
INT16> {<min y offset : INT16> <max y offset : INT16>}  
[<min theta : INT16> <max theta : INT16>]
```

Input References:

```
<position - sensor ID:INT16><position - output  
ID:INT16><low intensity - sensor ID:INT16> <low  
intensity - output ID:INT16>< high intensity - sensor  
ID: INT16> <high intensity - output ID : INT16>
```

Outputs:

```
<x : INT32> <y : INT32> <theta : INT32,INT32>  
<offset1 (%) : INT32> <offset2 (%) : INT32>  
<Min Intensity : INT32> <Max Intensity : INT32>  
<contrast : INT32> <Threshold : INT32>
```

Position during Learning:

```
<x : INT32> <y : INT32> <theta : INT32,INT32>
```

| Positioning Sensor Referencing IDs              |    |
|-------------------------------------------------|----|
| Input (i.e. – Will reference other sensors by): |    |
| Position                                        | 0  |
| Low Intensity Reference                         | 16 |
| High Intensity Reference                        | 17 |
| Output (i.e. – Can be referenced by)            |    |
| Position                                        | 0  |
| Min Intensity                                   | 16 |
| Max Intensity                                   | 17 |
| Contrast                                        | 20 |
| Threshold                                       | 21 |

| Failure Code | Warning Code | Positioning (Translational/Rotational) SoftSensors |
|--------------|--------------|----------------------------------------------------|
| -1           | 1            | value < minimum contrast                           |
| -2           | 2            | value < minimum percent offset1                    |
| -3           | 3            | value > maximum percent offset1                    |
| -4           | 4            | value < minimum percent offset2                    |
| -5           | 5            | value > maximum percent offset2                    |
| -6           | 6            | bad line fit (area edge)                           |
| -7           |              | no intersection (area edge)                        |
| -8           |              | no edge found (lines, arcs)                        |

## PrecisionMeasure Sensors (Module ID: 4)

### Command Format For Setting PrecisionMeasure Sensor Parameters

**SP** <sensor ID : INT16><parameter ID : UINT8><data>

<parameter ID> :

0 Set sensor type and coordinates

<data>  $\Rightarrow$  <type : UINT8><coord data>

<type>

1 straight line

<coord data>  $\Rightarrow$  <x1 : INT16> <y1 : INT16> <x2 : INT16> <y2 : INT16>

2 circular arc (toroid)

<coord data>  $\Rightarrow$  <xc : INT16> <yc : INT16> <xr1 : INT16> <xr2 : INT16>

4 area-edge sensor (line)

<coord data>  $\Rightarrow$  <x1 : INT16> <y1 : INT16> <x2 : INT16> <y2 : INT16> <x3 : INT16> <y3 : INT16>

12 area-edge sensor (circle) (defunct)

<coord data>  $\Rightarrow$  <x1 : INT16> <y1 : INT16> <x2 : INT16> <y2 : INT16> <x3 : INT16> <y3 : INT16>

17 straight line gauge

<coord data>  $\Rightarrow$  <x1 : INT16> <y1 : INT16> <x2 : INT16> <y2 : INT16>

20 area-edge gauge

<coord data>  $\Rightarrow$  <x1 : INT16> <y1 : INT16> <x2 : INT16> <y2 : INT16> <x3 : INT16> <y3 : INT16>

2 Set Parameters

<data>  $\Rightarrow$  <edge type : UINT4> Other parameters to follow

<edge type>

0 Positive or Negative gradient

1 Positive gradient

2 Negative gradient

if shape is line or outside-in line

<algorithm : UINT4> <scan density : INT32>

<algorithm>

for area-edge (line) and area-edge gauge sensors

0 Line Fit

1 Minimum Distance

2 Maximum Distance

3 Mean Distance

4 Median Distance

for circular arc and area-edge (circle) sensors

0 Circle Fit Radius

1 Minimum Radius

## 2 Maximum Radius

if shape is outside-in line or outside-in parallelogram  
    <Inside Out Flag: UINT8>

<Scale type : UINT4> <<data>>

    <Scale type>

    0        no scaling.

    1        reference scale.

        <<data>>  $\Rightarrow$  <sensorID : INT16>

<Stage 1 Type: UINT8> <Stage 1 Value : UINT8> <Stage 2 Type:  
UINT8> <Stage 2 Value: UINT8> <Flags: UINT32> <Outlier Distance: INT32>  
<Edge Width: UINT32> <Number of Iterations : INT32>

8    set sensor warning levels

    <data>  $\Rightarrow$  <min contrast : INT16> <min distance : FLOAT64>

        <max distance : FLOAT64>

    if shape is parallelogram or outside-in parallelogram

        <min angle : FLOAT64> <max angle : FLOAT64>

        <max straightness : FLOAT64>

    else if shape is torus or circle parallelogram

        <max roundness : FLOAT64>

        angle = (angle in degrees(-90..+90)) + 90

9    set sensor passing levels

    <data>  $\Rightarrow$  < same <data> as #8>

## **PrecisionMeasure Sensor Query Response**

*<type/shape : UINT8><<data>>*

- 0 no shape defined
- 1 straight line  
*<data> = <x1> <y1> <x2> <y2>*
- 2 circular toroid  
*<data> = <cx> <cy> <xr1> <xr2>*
- 4 area-edge locator (line)  
*<data> = <x1> <y1> <x2> <y2> <x3> <y3> <x4> <y4>*
- 12 area-edge locator (circle)  
*<data> = <x1> <y1> <x2> <y2> <x3> <y3> <x4> <y4>*
- 17 straight line gauge  
*<data> = <x1> <y1> <x2> <y2>*
- 20 area-edge gauge  
*<data> = <x1> <y1> <x2> <y2> <x3> <y3> <x4> <y4>*

*<threshold type : UINT4> <value : INT16>*

- 0 fixed gradient value
- 1 percent of contrast along path
- 2 percent of reference input contrast

*<edge type : UINT4>*

- 0 either
- 1 positive edge
- 2 negative edge

*<scaling type : UINT4> <<sensorID : INT16>>*

warning levels:

*<min contrast : INT16> <min distance/radius : FLOAT64>  
<max distance/radius : FLOAT64> <<min theta : FLOAT64>  
<max theta : FLOAT64>> <<max straightness : FLOAT64>>  
<<max roundness : FLOAT64>>*

failure levels:

same format as warning levels

Input References:

*<position - sensor ID:INT16><position - output  
ID:INT16><low intensity - sensor ID:INT16> <low  
intensity - output ID:INT16>< high intensity - sensor  
ID: INT16> <high intensity - output ID : INT16>*

Outputs:

```
<x : INT32> <y : INT32> <theta : INT32,INT32>
<<distance : FLOAT64>> <<radius : FLOAT64>>
<<angle : FLOAT64>> <Min Intensity : INT32>
<Max Intensity : INT32> <contrast : INT32>
<Gradient Threshold : INT32> <<Straightness : FLOAT64>>
<<Roundness : FLOAT64>> <scale factor : FLOAT64>
```

Position during Learning:

```
<x : INT32> <y : INT32> <theta : INT32,INT32>
```

| Measurement Sensor Referencing IDs              |     |
|-------------------------------------------------|-----|
| Input (i.e. – Will reference other sensors by): |     |
| Position                                        | 0   |
| Output (i.e. – Can be referenced by)            |     |
| Position                                        | 0   |
| Min Intensity                                   | 16  |
| Max Intensity                                   | 17  |
| Contrast                                        | 20  |
| Gradient Threshold                              | 21  |
| Subpixel Position                               | 96  |
| Subpixel Line                                   | 97  |
| Subpixel Circle                                 | 98  |
| Distance                                        | 112 |
| Angle                                           | 113 |
| Straightness                                    | 114 |
| Roundness                                       | 115 |
| Scale                                           | 116 |

| Failure Code | Warning Code | Measurement SoftSensors      |
|--------------|--------------|------------------------------|
| -1           | 1            | value < minimum contrast     |
| -2           | 2            | value < minimum distance     |
| -3           | 3            | value > maximum distance     |
| -4           | 4            | value < minimum radius/angle |
| -5           | 5            | value > maximum radius/angle |
| -6           | 6            | not enough edges found       |
| -7           |              | no intersection (area edge)  |
| -8           |              | straightness/roundness       |

## Math Sensors (Module ID: 5)

### Command Format For Setting Math Sensor Parameters

**SP**<sensor ID : INT16><parameter ID : UINT8><data>

<parameter ID> :

0 Set sensor type and coordinates

<data> ⇒ <type : UINT8><reference data>

<type>

1 distance calculator

<reference data> ⇒ <sensor ID1 : INT16> <param ID1 : INT16>  
<sensor ID2 : INT16> <param ID2 : INT16>

2 angle calculator

<reference data> ⇒ <sensor ID1 : INT16> <param ID1 : INT16>  
<sensor ID2 : INT16> <param ID2 : INT16>

3 intersection calculator

<reference data> ⇒ <sensor ID1 : INT16> <param ID1 : INT16>  
<sensor ID2 : INT16> <param ID2 : INT16>

4 midpoint calculator

<reference data> ⇒ <sensor ID1 : INT16> <param ID1 : INT16>  
<sensor ID2 : INT16> <param ID2 : INT16>

5 midline calculator

<reference data> ⇒ <sensor ID1 : INT16> <param ID1 : INT16>  
<sensor ID2 : INT16> <param ID2 : INT16>

6 line through two points calculator

<reference data> ⇒ <sensor ID1 : INT16> <param ID1 : INT16>  
<sensor ID2 : INT16> <param ID2 : INT16>

7 perpendicular line calculator

<reference data> ⇒ <sensor ID1 : INT16> <param ID1 : INT16>  
<sensor ID2 : INT16> <param ID2 : INT16>

8 scale factor calculator

9 coordinate transform

<reference data> ⇒ <Image Pt0 X : FLOAT64> < Image Pt0  
Y: FLOAT64> < Image Pt1 X : FLOAT64> < Image Pt1  
Y : FLOAT64> <World Pt0 X : FLOAT64> < World Pt0  
Y: FLOAT64> < World Pt1 X : FLOAT64> < World Pt1  
Y : FLOAT64> < Negative Z : UINT8> < Angle  
Scale : FLOAT64> < Angle Offset : FLOAT64>  
<sensor ID1 : INT16> <param ID1 : INT16>

10 coordinate system

<reference data> ⇒ <NumPoints : INT16>  
<CalculationFlags : UINT32> <<DistortionCoef : FLOAT64>>

11 new coordinate transform

<reference data> ⇒ <sensor ID1 : INT16> <param ID1 : INT16>  
<sensor ID2 : INT16> <param ID2 : INT16>

3 set angle range (for angle math sensor)

<data> ⇒ <type : UINT4>

```

<type>
0      -180 .. +180 degrees
1      0 .. 360 degrees
2      -90 .. +90 degrees
3      0 .. 180 degrees

```

3 set coordinate data (for coordinate system math sensor)

```

<data> ⇒ <PointIndex : INT16> <SensorRefID : INT16> <Image X : FLOAT64>
<Image Y : FLOAT64> <World X : FLOAT64> <World Y : FLOAT64>

```

Note: After setting the last point data, the coordinate system is recalculated.

4 set scale factor params for distance SoftSensor

```

<data> ⇒ <type : UINT4> <<data>>

```

```

<type>
0      no scaling.
1      reference scale.
<<data>> ⇒ <sensorID : INT16>

```

5 set scale factor params for Scale SoftSensor.

```

<data> ⇒ <type : UINT4> <<data>> <precision : INT8>

```

```

<type>
0      static scale.
<<data>> ⇒ <scale : FLOAT64> <sensorID : INT16>
<desired value : FLOAT64> <precision : INT8>
<units : string 19>
1      dynamic scale.
<<data>> ⇒ <sensorID : INT16> <desired value : FLOAT64>
<precision : INT8> <units : string 19>

```

8 set sensor warning levels

```

<data> ⇒ <min distance/angle : FLOAT64>
<max distance/angle : FLOAT64>

```

9 set sensor passing levels

```

<data> ⇒ < same <data> as #8>

```

## Math Sensor Query Response

```
<type/shape : UINT8> <newline> <<data>>
0    no shape defined
1    distance calculator
    <data> = <sensor ID1 : INT16> <param ID1 : INT16>
        <sensor ID2 : INT16> <param ID2 : INT16>
2    angle calculator
    <data> = <sensor ID1 : INT16> <param ID1 : INT16>
        <sensor ID2 : INT16> <param ID2 : INT16>
3    intersection calculator
    <data> = <sensor ID1 : INT16> <param ID1 : INT16>
        <sensor ID2 : INT16> <param ID2 : INT16>
4    midpoint calculator
    <data> = <sensor ID1 : INT16> <param ID1 : INT16>
        <sensor ID2 : INT16> <param ID2 : INT16>
5    midline calculator
    <data> = <sensor ID1 : INT16> <param ID1 : INT16>
        <sensor ID2 : INT16> <param ID2 : INT16>
6    line through two points calculator
    <data> = <sensor ID1 : INT16> <param ID1 : INT16>
        <sensor ID2 : INT16> <param ID2 : INT16>
7    perpendicular line calculator
    <data> = <sensor ID1 : INT16> <param ID1 : INT16>
        <sensor ID2 : INT16> <param ID2 : INT16>
8    scale factor calculator
    <data> = <ScaleType : UINT4> <<Scale : FLOAT64>>
        <Sensor ID : INT16> <DesiredValue : FLOAT64>
        <Precision : INT8> <Units : string 19>
9    coordinate transformation
    <data> = <Image Pt0 X : FLOAT64> < Image Pt0 Y: FLOAT64> <
        Image Pt1 X : FLOAT64> < Image Pt1 Y : FLOAT64> <World
        Pt0 X : FLOAT64> < World Pt0 Y: FLOAT64> < World Pt1
        X : FLOAT64> < World Pt1 Y : FLOAT64> < Negative
        Z : UINT8> < Angle Scale : FLOAT64> < Angle
        Offset : FLOAT64> <sensor ID1 : INT16>
        <param ID1 : INT16>
10   coordinate system
    <data> = <NumPoints : INT16> <CalculationFlags : UINT32>

    <0 : INT16> <SensorID : INT16> <Image X : FLOAT64>
        <Image Y : FLOAT64> <World X : FLOAT64>
        <World Y : FLOAT64>
    ...
    <NumPoints-1 : INT16> <SensorID : INT16>
        <Image X : FLOAT64> <Image Y : FLOAT64>
        <World X : FLOAT64> <World Y : FLOAT64>

    <T0 : FLOAT64> <T1 : FLOAT64> ... <T11 : FLOAT64>
        <ImageCenterX : FLOAT64> <ImageCenterY : FLOAT64>
        <DistortionCoef : FLOAT64> <FocalLength : FLOAT64>
11   new coordinate transformation
```

<data> = <sensor ID1 : INT16> <param ID1 : INT16>  
 <sensor ID2 : INT16> <param ID2 : INT16>

angle range:

<angle range : UINT4>

Scaling Params (for distance SoftSensor):

<<type : UINT4> <<sensorID : INT16>>

warning levels:

<<min distance/angle : FLOAT64> <max distance/angle :  
 FLOAT64>>

failure levels:

<<min distance/angle : FLOAT64> <max distance/angle :  
 FLOAT64>>

Outputs:

<<distance : FLOAT64> <scale : FLOAT64>>  
 <<angle : FLOAT64>> <<x : FLOAT64> <y : FLOAT64>>  
 <<x0 : FLOAT64> <y0 : FLOAT64> <a : FLOAT64>  
 <b : FLOAT64>> <<scale factor : FLOAT64>> <<x : FLOAT64>  
 <y : FLOAT64> <angle : FLOAT64>>

| Math Sensor Referencing IDs                     |     |
|-------------------------------------------------|-----|
| Input (i.e. – Will reference other sensors by): |     |
| Distance                                        | 112 |
| Scale                                           | 116 |
| Output (i.e. – Can be referenced by)            |     |
| Position                                        | 0   |
| Subpixel Position                               | 96  |
| Subpixel Line Position                          | 97  |
| Distance                                        | 112 |
| Angle                                           | 113 |
| Scale                                           | 116 |

| Failure Code | Warning Code | Math SoftSensors       |
|--------------|--------------|------------------------|
| -1           | 1            | minimum distance/angle |
| -2           | 2            | maximum distance/angle |

## Math (Coordinate System/Type==10) Sensor Query:

Look at Sensor Query command (SQ<SensorId:INT16>) in "Sensor Management Commands" section. The "sensor specific information" for a Math Sensor/Coordinate System is shown below.

< ... Sensor Specific Information ... > =

<type/shape==10 : UINT8>

<NumPoints : INT16> <Flags : UINT32>

<index : INT16> <SensorRef : INT16> <ImageX : FLOAT64> <ImageY :  
FLOAT64> <UserX : FLOAT64> <UserY : FLOAT64>

...

repeat for each point in calibration set (NumPoints of data)

<r1 : FLOAT64> ... <r3 : FLOAT64> <r4 : FLOAT64> ... <r6 : FLOAT64> <Tx :  
FLOAT64> <r7 : FLOAT64> ... <r9 : FLOAT64> <Tz : FLOAT64> <CenterX  
: Float64> <CenterY : Float64> < Distortion : FLOAT64> < Focal  
Length : FLOAT64>

## Command Format For Setting Math(Coordinate System/Type==10) Sensor Parameters

Set the sensor type and basic data.

**SP**<sensorId : INT16> <parameter ID==0 : UINT8> <type==10 : UINT8>  
<NumPoints : INT16> <Flags : UINT32> <<Distortion : FLOAT64>>

Set the Coordinate System Calibration data.

**SP**<sensorId : INT16> <parameter ID==3 : UINT8> <Index : INT16>INT8>  
<SensorRef : INT16> <ImageX : FLOAT64> <ImageY : FLOAT64> <UserX :  
FLOAT64> <UserY : FLOAT64>  
... repeat for each calibration point (NumPoints)

Flags is a bitfield,

| Bit | Description                                          |
|-----|------------------------------------------------------|
| 0   | Set to enable calculation of transformation matrix.  |
| 1   | Set to enable distortion coefficient re-calculation. |
| 2   | Set for "Aligned Z axis" mode.                       |
| 3   | Set for "Negative Z axis" mode.                      |
| 4   | Link to Global Calibration                           |

As an example:

| Command/Response | Description                                 |
|------------------|---------------------------------------------|
| #SQ              | Query list of sensors                       |
| \$ 3 5 Sensor3   | Sensor ID: 3, Type: 5 (Math), Name: Sensor3 |

|                                              |                                                                                                |
|----------------------------------------------|------------------------------------------------------------------------------------------------|
| % 0                                          | Command completed without error.                                                               |
| ?                                            | Command completed without error.                                                               |
| #SQ3                                         | Query sensor 3 for its parameters                                                              |
| \$ 3 1 Sensor3                               |                                                                                                |
| \$ 0 0 0 0 0 0                               |                                                                                                |
| \$                                           | Sensor Description                                                                             |
| \$ 10                                        | Start sensor spec. data (Math, Type: 10/Coord System)                                          |
| \$ 1 13                                      | 1 Data point, Flags = 13 (\$0d) (Neg, aligned z axis, enable translation/rotation calibration) |
| \$ 0 -1 1 2 3 4                              | Data Point 0: Ref=-1, Image=(1.0,2.0), User=(3.0,4.0)                                          |
| \$ 1 0 0 -320 0 1 0 -240 0 0 1 1 320 240 0 1 | Calibration results:                                                                           |
|                                              | 1.0    0    0    -320                                                                          |
|                                              | 0    1.0    0    -240                                                                          |
|                                              | 0    0    1.0    1                                                                             |
|                                              | Image center: (320,240), Distortion: 0, Focal Length: 1.0                                      |
| % 0                                          | Command completed without error.                                                               |
| ?                                            | Command completed without error.                                                               |
| #SP3 0 10 1 13 (0)                           | Set sensor type==10, 1 data point, flags==13, distortion==0                                    |
| % 0                                          | Command completed without error.                                                               |
| ?                                            | Command completed without error.                                                               |
| #SP3 3 0 -1 1 2 3 4                          | Set data point 0: ref=-1, image coord=(1,2), user coord=(3,4)                                  |
| % 0                                          | Command completed without error.                                                               |
| ?                                            | Command completed without error.                                                               |

## Data Matrix SoftSensor (Module ID: 6)

### Command Format For Setting Data Matrix Sensor Parameters

SP <sensor ID : INT16><parameter ID : UINT8><data>

<parameter ID>:

2 set code parameters

<data>  $\Rightarrow$  <edge type : UINT8> <ECC Type: UINT8>  
<symbol size : UINT8> <max Skew : UINT8>  
<min cell size : INT16> <max cell size : INT16>  
<fill density : UINT8> <Mirror Image : UINT8> <Border Integrity: UINT8> <max Lines Used: UINT8> <Marking Mode: INT16> <Marking Index: INT16> <Flags: UINT32> <QZ Integrity: UINT8> <Axial Nonuniformity: UINT8> <AngleBins : INT32> <DistanceBins : INT32> <LineStraightness : INT32>

edge type:

- 0 Dark on Light Code
- 1 Light on Dark Code
- 2 Autodetect

symbol size (for ECC200):

- |    |              |    |              |    |              |    |              |
|----|--------------|----|--------------|----|--------------|----|--------------|
| 0  | SIZE_10_10   | 1  | SIZE_12_12   | 2  | SIZE_14_14   | 3  | SIZE_16_16   |
| 4  | SIZE_18_18   | 5  | SIZE_20_20   | 6  | SIZE_22_22   | 7  | SIZE_24_24   |
| 8  | SIZE_26_26   | 9  | SIZE_32_32   | 10 | SIZE_36_36   | 11 | SIZE_40_40   |
| 12 | SIZE_44_44   | 13 | SIZE_48_48   | 14 | SIZE_52_52   | 15 | SIZE_64_64   |
| 16 | SIZE_72_72   | 17 | SIZE_80_80   | 18 | SIZE_88_88   | 19 | SIZE_96_96   |
| 20 | SIZE_104_104 | 21 | SIZE_120_120 | 22 | SIZE_132_132 | 23 | SIZE_144_144 |
| 24 | SIZE_8_18    | 25 | SIZE_8_32    | 26 | SIZE_12_26   | 27 | SIZE_12_36   |
| 28 | SIZE_16_36   | 29 | SIZE_16_48   |    |              |    |              |

max skew:

angle in degrees: 0 - 45

min/max cell size (width):

(size in pixels)\*100: 2\*100 – 100\*100

fill density:

- 0 > 30%
- 1  $\leq$  30%
- 2 both

Mirror Image:

- 0: normal symbol
- 1: mirror image symbol
- 2: autodetect

Flags:

- Bit 0: Use digital relearn
- Bit 1: Mark Intensity
- Bit 2: Advanced Search

- 3 Set Preprocessing
  - <data> ⇒ <Pre Processing Type 1 : UINT8> <Pre Processing Width 1 : UINT16> <Pre Processing Height 1 : UINT16> <Pre Processing Type 2 : UINT8> <Pre Processing Width 2 : UINT16> <Pre Processing Height 2 : UINT16>*
- 8 Set Minimum Contrast
  - <data> ⇒ <Minimum Contrast : INT32>*
- 9 Set Pass Criteria
  - <data> ⇒ <start location : INT16> <string : string 200>*  
*if start location = -1, then match entire string, otherwise match substring starting at start location.*

## Data Matrix Sensor Query Response

<Base Position Reference Sensor Data>

<Shape Data>

<edge type : UINT8> <ECC Type: UINT8> <symbol size : UINT8>  
<max Skew : UINT8> <min cell size : INT16>  
<max cell size : INT16> <fill density : UINT8> <Mirror Image :  
UINT8> <Border Integrity: UINT8> <max Lines Used: UINT8> <Marking  
Mode: INT16> <Marking Index: INT16> <Flags: UINT32> <QZ  
Integrity: UINT8> <Axial Nonuniformity: UINT8>  
<AngleBins : INT32> <DistanceBins : INT32>  
<LineStraightness : INT32>  
<Pre Processing Type 1 : UINT8> <Pre Processing Width 1 : UINT16>  
<Pre Processing Height 1 : UINT16> <Pre Processing Type 2 :  
UINT8> <Pre Processing Width 2 : UINT16> <Pre Processing Height 2  
: UINT16>  
<Minimum Contrast : INT32>  
<Debug1 : INT32> <Debug2 : INT32> <Debug3 : INT32> <Debug4 : INT32>  
<start location : INT16> <string : string 200>

Pass Criteria:

<start location : INT16> <string : string 200>

Input References:

<position - sensor ID:INT16><position - output ID:INT16>

Outputs:

<string : string 200>

Position during Learning:

<x : INT32> <y : INT32> <theta : INT32,INT32>

| Data Matrix Sensor Referencing IDs              |     |
|-------------------------------------------------|-----|
| Input (i.e. – Will reference other sensors by): |     |
| Position                                        | 0   |
| Output (i.e. – Can be referenced by)            |     |
| String                                          | 128 |

### Format of encoded string:

A '~' is an escape character. Any non-printing character is printed as a '~xx', where xx is the ASCII code, in hex. A '~7e' is used to represent a '~' in the data string.

| Failure Code | Warning Code | DataMatrix SoftSensors              |
|--------------|--------------|-------------------------------------|
| -1           | 1            | String does not match pass criteria |

## Blob Generation SoftSensor (Module ID: 7)

### Command Format For Setting Blob Generation Sensor Parameters

**SP** <sensor ID : INT16><parameter ID : UINT8><data>

<parameter ID>:

- 0 Set sensor shape/type  
<data> ⇒ <type : UINT8><coord data>  
  
<type>  
3 rectangle  
<coord data> ⇒ <x0 : INT16> <y0 : INT16> <x1 : INT16>  
<y1 : INT16>
- 2 Set general information  
<data> ⇒ <Connectivity Type : UINT4> <Use Boundary Blobs : UINT4> <Min Blob Size : INT32> <Max Blob Size : INT32>  
Connectivity type:  
0 4-Connectedness  
1 8-Connectedness
- 3 Set preprocessing information  
<data> ⇒ <Filter Type: UINT4> <Binarization Type : UINT4>  
<Preproc. 1 Type : INT8> <Preproc. 1 Size : INT32> ...  
<Preproc. 5 Type : INT8> <Preproc. 5 Size : INT32>  
<Maximum Connected Component Blobs: INT32>  
Filter Type:  
0 Edges: None  
1 Edges: Roberts Operator  
2 Edges: Prewitt Operator  
3 Edges: Sobel Operator  
Binarization type:  
0 Dark Blobs  
1 Light Blobs  
Preprocessing type:  
0 None  
1 Dilation  
2 Erosion  
3 Open  
4 Close
- 8,9 Set Warn/Pass Criteria  
<data> ⇒ <Min Contrast : INT16> <Min Number Blobs : INT16>  
<Max Number Blobs : INT16>

### **Blob Generation Sensor Query Response**

<type/shape : UINT8><<data>>

0 no shape defined  
3 rectangle  
    <data> = <x0> <y0> <x1> <y1>

<threshold type : UINT4> <value : UINT8>

0 Fixed greyscale value  
1 Percent of contrast along path  
2 Percent of reference input contrast

<Connectivity Type : UINT4> <Use Boundary Blobs : UINT4> <Min Blob Size : INT32> <Max Blob Size : INT32>

<Binarization Type : UINT4> <Preproc. 1 Type : INT8> <Preproc. 1 Size : INT32> ... <Preproc. 6 Type : INT8> <Preproc. 6 Size : INT32>

Warn Criteria:

<Min Contrast : INT16> <Min Num Blobs : INT16> <Max Num Blobs : INT16>

Pass Criteria:

<Min Contrast : INT16> <Min Num Blobs : INT16> <Max Num Blobs : INT16>

Input References:

<position - sensor ID:INT16><position - output ID:INT16>  
<Min Intensity - sensor ID:INT16>< Min Intensity - output ID:INT16>  
< Max Intensity - sensor ID:INT16>< Max Intensity - output ID:INT16>

Outputs:

< Min Intensity :INT32>< Max Intensity :INT32> < Contrast :INT32> < Threshold :INT32> < Num Blobs :INT32>

Position during Learning:

<x : INT32> <y : INT32> <theta : INT32,INT32>

| <b>Blob Generation Sensor Referencing IDs</b>   |     |
|-------------------------------------------------|-----|
| Input (i.e. – Will reference other sensors by): |     |
| Position                                        | 0   |
| Low Intensity Reference                         | 16  |
| High Intensity Reference                        | 17  |
| Output (i.e. – Can be referenced by)            |     |
| Min Intensity                                   | 16  |
| Max Intensity                                   | 17  |
| Contrast                                        | 20  |
| Threshold                                       | 21  |
| Number of Blobs                                 | 35  |
| Blob Feature Vector                             | 144 |

| <b>Failure Code</b> | <b>Warning Code</b> | <b>Blob Generation SoftSensors</b> |
|---------------------|---------------------|------------------------------------|
| -1                  | 1                   | Minimum contrast                   |
| -2                  | 2                   | Minimum blobs                      |
| -3                  | 3                   | Maximum blobs                      |

## Blob Selection SoftSensor (Module ID: 8)

### Command Format For Setting Blob Selection Sensor Parameters

**SP** <sensor ID : INT16><parameter ID : UINT8><data>

<parameter ID>:

- 0 Set sensor shape/type  
<data>  $\Rightarrow$  <type : UINT8><coord data>  
  
<type>
  - 0 Group selection
- 2 General Blob Selector parameters  
<data>  $\Rightarrow$  <angle type : INT16> <blob reference : INT16>  
angle type:
  - 0 Principle Axis ( $\pm 180^\circ$ )
  - 1 Principle Axis (0-360 $^\circ$ )
  - 2 Maximum Point from Centroid
- 3 Group algorithm parameters  
<data>  $\Rightarrow$  <Min Area : INT32> <Max Area : INT32> <Min Perimeter : FLOAT64> <Max Perimeter : FLOAT64> <Min Position - X : FLOAT64> <Min Position - Y : FLOAT64> <Max Position - X : FLOAT64> <Max Position - Y : FLOAT64> <Min Angle : FLOAT64> <Max Angle : FLOAT64> <Min Intensity : INT32> <Max Intensity : INT32> <Min Eccentricity : FLOAT64> <Max Eccentricity : FLOAT64> <Min Compactness : FLOAT64> <Max Compactness : FLOAT64> <Min Bounding Box Width : INT32> <Max Bounding Box Width : INT32> <Min Bounding Box Height : INT32> <Max Bounding Box Height : INT32> <Min Radius : FLOAT64> <Max Radius : FLOAT64>
- 8,9 Set Warn/Pass Criteria  
<data>  $\Rightarrow$  <Minimum Number Blobs : INT32> <Maximum Number Blobs : INT32>
- 10 Set input reference  
<data>  $\Rightarrow$  <Blob Generation Sensor ID : INT16>

### **Blob Selection Sensor Query Response**

<type/shape : UINT8><<data>>

0 Group selection algorithm

<angle type : INT16> <blob reference : INT16>

<Min Area : INT32> <Max Area : INT32> <Min Perimeter : FLOAT64> <Max Perimeter : FLOAT64> <Min Position - X : FLOAT64> <Min Position - Y : FLOAT64> <Max Position - X : FLOAT64> <Max Position - Y : FLOAT64> <Min Angle : FLOAT64> <Max Angle : FLOAT64> <Min Intensity : INT32> <Max Intensity : INT32> <Min Eccentricity : FLOAT64> <Max Eccentricity : FLOAT64> <Min Compactness : FLOAT64> <Max Compactness : FLOAT64> <Min Bounding Box Width : INT32> <Max Bounding Box Width : INT32> <Min Bounding Box Height : INT32> <Max Bounding Box Height : INT32> <Min Radius : FLOAT64> <Max Radius : FLOAT64>

Warn/Pass Criteria:

<Minimum Number Blobs : INT32> <Maximum Number Blobs : INT32>

Input References:

<Blob Generation Sensor ID : INT16>

Outputs:

<x : INT32> <y : INT32> <theta : INT32,INT32> <Number of Blobs : INT32>

| Blob Selection Sensor Referencing IDs           |         |
|-------------------------------------------------|---------|
| Input (i.e. – Will reference other sensors by): |         |
| Blob Feature Vector                             | 144     |
| Output (i.e. – Can be referenced by)            |         |
| Position                                        | 0       |
| Number of Blobs                                 | 35      |
| Subpixel Position                               | 96      |
| Blob Area (Blobs 1-8)                           | 256-263 |
| Blob Subpixel Position (Blobs 1-8)              | 272-279 |
| Blob Angle (Blobs 1-8)                          | 288-295 |

| Failure Code | Warning Code | Blob Selection SoftSensors |
|--------------|--------------|----------------------------|
| -1           | 1            | Minimum blobs              |
| -2           | 2            | Maximum blobs              |

## Scripting SoftSensor (Module ID: 9)

### Command Format For Setting Scripting Sensor Parameters

**SP**<sensor ID : INT16><parameter ID : UINT8><data>

<parameter ID> :

- 0 Set general sensor parameters  
<data> ⇒ <Program Length : UINT16> <Num Tokens : UINT16> <Stack Depth : UINT16>  
<Output Bits : UINT16>
- 1 Add a program line  
<data> ⇒ <Number of Line Characters : UINT8> <Line Char 1 : UINT8>  
... <Line Char n : UINT8>

NOTE: Line Characters are transmitted as their ASCII values.

- 2 Add program tokens  
<data> ⇒ <Number of Tokens sent : INT16> <Token  
1 : INT16><<token data>> ... <Token n : INT16><<token data>>

| token                 | <token data>                                 |
|-----------------------|----------------------------------------------|
| TOKEN_PUSH1(1)        | <i : INT32>                                  |
| TOKEN_PUSHF(2)        | <d : FLOAT64>                                |
| TOKEN_PUSHSTR (12)    | not implimented                              |
| TOKEN_JUMP_FALSE (3)  | <jump addr : INT32>                          |
| TOKEN_JUMP (4)        |                                              |
| TOKEN_LOAD (5)        | <id : INT32>                                 |
| TOKEN_STORE (6)       |                                              |
| TOKEN_FUNCTION (8)    |                                              |
| TOKEN_LOAD_SREF (9)   | <Sid : INT16> <Oid : INT16><br><Mid : INT16> |
| TOKEN_CHECK_SREF (11) |                                              |
| TOKEN_STORE_SREF (10) | <Oid : INT16> <Mid : INT16>                  |

- 3 Set Warn string  
<data> ⇒ <Warn String : string 200>
- 4 Set Fail string  
<data> ⇒ <Fail String : string 200>

## **Scripting Sensor Query Response**

<type : UINT8> <Program Length : UINT16> <Number of Tokens : UINT16>  
    <Stack Depth : UINT16> <Output Bits : UINT16>

<Number Program Line Characters : UINT8> <Line Char 1 : UINT8> ... <Line  
Char n : UINT8>

.  
.  
.

<Number of Tokens Sent : UINT8> <Token 1 : UINT8> <<Token Data >> ...  
    <Token n : UINT8> <<Token Data >>

    <<Token Data >> See above

.  
.  
.

<Warn String : STRING200>

<Fail String : STRING200>

### **Outputs:**

<x : INT32> <y : INT32> <theta : INT32,INT32> <Intensity : INT32>  
    <Distance : FLOAT64> <Angle : FLOAT64> <Subpixel X : FLOAT64>  
    <Subpixel Y : FLOAT64> <Line Xo: FLOAT64> <Line Yo : FLOAT64> <Line  
a : FLOAT64> <Line b : FLOAT64>

| <b>Scripting Sensor Referencing IDs</b>         |         |
|-------------------------------------------------|---------|
| Input (i.e. – Will reference other sensors by): |         |
| Position                                        | 0       |
| Minimum Intensity                               | 16      |
| Maximum Intensity                               | 17      |
| Mean Intensity                                  | 18      |
| Median Intensity                                | 19      |
| Contrast                                        | 20      |
| Threshold                                       | 21      |
| # of Edges                                      | 32      |
| # of Features                                   | 33      |
| # of Pixels                                     | 34      |
| # of Blobs                                      | 35      |
| % below threshold                               | 48      |
| Subpixel Position                               | 96      |
| Subpixel Line                                   | 97      |
| Subpixel Circle                                 | 98      |
| Distance/Radius                                 | 112     |
| Angle                                           | 113     |
| Straightness                                    | 114     |
| Roundness                                       | 115     |
| Bar Code String                                 | 128     |
| Blob Area (Blobs 1-8)                           | 256-263 |
| Blob Subpixel Position (Blobs 1-8)              | 272-279 |
| Blob Angle (Blobs 1-8)                          | 288-295 |
| Output (i.e. – Can be referenced by)            |         |
| Position                                        | 0       |
| Intensity                                       | 16      |
| Subpixel Position                               | 96      |
| Subpixel Line                                   | 97      |
| Distance                                        | 112     |
| Angle                                           | 113     |
| String                                          | 128     |

| <b>Failure Code</b> | <b>Warning Code</b> | <b>Scripting SoftSensors</b>             |
|---------------------|---------------------|------------------------------------------|
| -1                  | 1                   | Result set to FAIL or WARN               |
| -2                  | 2                   | Processing error                         |
| <-2                 | >2                  | Result set to unknown fail or warn value |

## OCR SoftSensors (Module ID: 13)

### Command Format For Setting OCR Sensor Parameters

**SP** <sensor ID : INT16><parameter ID : UINT8><data>

<parameter ID> :

- 0 Set sensor shape/type
- 2 <Filter Type: UINT8> <Filter Size: UINT8> <Filter Clipping: UINT16>
- 4 Start Font List Transfer  
<Font File Size (Bytes): UINT32>  
NOTE: When this command is received the current font list is deleted and all font list character counters are reset to zero.
- 5 Font List Data Block Transfer  
<Font List Data: UINT8>...  
NOTE: This command is called repeatedly until the entire Font List data has been transferred (the number of bytes sent must match the value sent with parameter ID 4).  
WARNING: Parameter IDs 4 and 5 are not intended to be used by users as the data is raw binary values representing an unspecified list of data structures.
- 6 OCR Library Parameters  
<Text Color: UINT8> <Horizontal Spacing: UINT8> <Vertical Spacing: UINT8> <Use Aspect Ratio Range : UINT8> <Use Relaxed Mode : UINT8> <Matching Resolution : UINT8> <Acceptance Percentage :UINT16> <Minimum Character Size (Pixels) : UINT32> <Minimum Spot Size : UINT32> <Substitution Character (ASCII value) : UINT8> <Use Maximum Character Width : UINT4> <Use Advanced Maximum Character Width : UINT4> <Maximum Character Width Value : INT32> <Use Minimum Gap :UINT4> <Minimum Gap Value :UINT32> <Scan Type : UINT8> <Maximum Number of Characters Expected : INT32> <Aspect Ratio Range Value :UINT16> <Use Minimum Edge Width : UINT8> <Minimum Edge Width Value : UINT16> <Use Character Fill : UINT8> <Use Orphan Removal : UINT8> <Enable AutoFielding : UINT8> <Enable Auto Parameter Scaling on Rotation (only for Parallelogram shape) : UINT8>  
  
Text Color:  
Black - 0, White - 1  
Matching Resolution:  
High Resolution - 0, Low Resolution - 1, Medium Resolution - 3
- 7 Fielding Strings  
<Fielding String Index: INT32> <<Fielding String Value: string 100>>  
Fielding String Index:
  - 1 Clear all fielding strings
  - 0 - 49 Index of fielding string (character position) to copy Fielding String Value into

- 8 Train Characters (Add to Font List)  
 <Training Mode: UINT8> <Training String: string 50>  
 Training Mode: Not used at this time  
 NOTE: Training in this case referring to adding characters to the font list. Training will only occur during the next background (SM\_MARK\_MODE) inspection. This type of inspection will not occur unless a GUI interface (Framework, SmartLink, or ActiveX) is connected to the system.
- 9 Warn Values  
 <Minimum Contrast: INT16> <Minimum Score: INT16>
- 10 Pass Values  
 <Minimum Contrast: INT16> <Number of Characters Read: INT16>  
 <Fail if any Substitution Characters are Found: INT8> <Enable Match String: INT8> <Enable Digital Relearn of Match String: INT8> <Minimum Score: INT16>
- 12 Match String Relearn  
 <Match String: string 50>  
 NOTE: If Auto Fielding is enabled the values for the fielding strings will be changed to the same as the Match String at the time this command is issued.
- 14 Training Parameters  
 <Training Threshold Type: INT32> <Training Acceptance Threshold Value: INT16> <Font List Description Length (Number of Characters): INT8> <Font List Description: string 400>  
 Training Threshold Type:  
     1      Use Matching Acceptance Threshold  
     1      Use Training Acceptance Threshold  
     2      Train all characters regardless of match
- 15 Reset Font List Counters

## **OCR Sensor Query Response**

<Position Reference Sensor ID : INT16> <Position Reference Parameter ID : INT16>

<Learned Position X : INT32> <Learned Position Y : INT32> <Learned Position Theta X : INT32> <Learned Position Theta Y : INT32>

< Shape Information >

<Filter Type: UINT8> <Filter Size: UINT8> <Filter Clipping: UINT16>

<Text Color: UINT8> <Horizontal Spacing: UINT8> <Vertical Spacing: UINT8> <Use Aspect Ratio Range : UINT8> <Use Relaxed Mode : UINT8> <Matching Resolution : UINT8> <Acceptance Percentage :UINT16> <Minimum Character Size (Pixels) : UINT32> <Minimum Spot Size : UINT32> <Substitution Character (ASCII value) : UINT8> <Use Maximum Character

Width : UINT4> <Use Advanced Maximum Character Width : UINT4> <Maximum Character Width Value : INT32> <Use Minimum Gap :UINT4> <Minimum Gap Value :UINT32> <Scan Type : UINT8> <Maximum Number of Characters Expected : INT32> <Aspect Ratio Range Value :UINT16> <Use Minimum Edge Width : UINT8> <Minimum Edge Width Value : UINT16> <Use Character Fill : UINT8> <Use Orphan Removal : UINT8> <Enable AutoFielding : UINT8> <Enable Auto Parameter Scaling on Rotation (only for Parallelogram shape) : UINT8>

<0: INT16> <Fielding String 0: string 100>

.  
.  
.

<49: INT16> <Fielding String 49: string 100>

<Training Threshold Type: INT32> <Training Acceptance Threshold Value: INT16> <Font List Description Length (Number of Characters): UINT16> <Font List Description: string 400>

<Minimum Warn Contrast: INT16> <Minimum Warn Score: INT16>

<Minimum Pass Contrast: INT16> < Pass Number of Characters Read: INT16> <Fail if any Substitution Characters are Found: INT8> <Enable Pass Match String: INT8> <Enable Pass Digital Relearn of Match String: INT8> <Minimum Pass Score: INT16> <Pass Match String: string 50>

<Font File Size (Bytes): UINT32> <Font List Data: UINT8>...

<Font List Data: UINT8>...

.  
.  
.

...<Font List Data: UINT8>

#### Outputs:

<Minimum Intensity : INT32> <Maximum Intensity : INT32>  
<Contrast : INT32 > <Threshold : INT32> <OCR String : string>

| <b>OCR Sensor Referencing IDs</b>                      |     |
|--------------------------------------------------------|-----|
| <b>Input (i.e. – Will reference other sensors by):</b> |     |
| Position                                               | 0   |
| Minimum Intensity                                      | 16  |
| Maximum Intensity                                      | 17  |
| Mean Intensity                                         | 18  |
| Median Intensity                                       | 19  |
| Contrast                                               | 20  |
| Threshold                                              | 21  |
| <b>Output (i.e. – Can be referenced by)</b>            |     |
| Position                                               | 0   |
| Number of Characters                                   | 35  |
| Character Match Score                                  | 70  |
| OCR String                                             | 128 |
| Character Threshold                                    | 515 |
| Character Width                                        | 540 |
| Character Height                                       | 541 |
| Character Position                                     | 550 |
| Character Distance                                     | 551 |

| <b>Failure Code</b> | <b>Warning Code</b> | <b>OCR SoftSensors</b>           |
|---------------------|---------------------|----------------------------------|
| -1                  | 1                   | Minimum Contrast                 |
| -2                  | 2                   | Wrong Number of Characters       |
| -3                  | 3                   | Not All Characters Could Be Read |
| -4                  | 4                   | String Did Not Match             |
| -5                  | 5                   | Invalid Threshold                |
| -6                  | 6                   | Sensor Timeout                   |
| -7                  | 7                   | Sensor Locked                    |
| -8                  | 8                   | Sensor Not Read From Flash       |
| -9                  | 9                   | Invalid System                   |
| -10                 | 10                  | Invalid Parameter                |
| -11                 | 11                  | Minimum Character Score          |

## Vericode SoftSensors (Module ID: 14)

### Command Format For Setting Vericode Sensor Parameters

**SP** <sensor ID : INT16><parameter ID : UINT8><data>

<parameter ID> :

0 Set sensor shape/type

2 <Is Matrix Fixed: INT16> <Fixed Matrix Size: INT16> <Index Contrast: INT16> <Is Contrast Normal: INT16> <Edac Level: INT16> <Compression: INT16> <Symbol Type: INT16> <Bits Per Cell: INT16> <Sample Width: INT16> <Noise: INT16> <Pre-Filtering: INT16> <Filter Size: INT16> <Filter Iterations: INT16> <Use Packet: INT16> <Display Extracted Points: INT16> <Display Object Map: INT16> <Display Borders: INT16> <Display Vericode: INT16>

Index Contrast:

0 - Auto  
1 - 5%  
19 - 95%

Edac Level:

0 - Auto  
1 - 12.5%  
2 - 25%

Compression:

0 - 8 bit  
-1 - 4 bit  
-2 - 6 bit

Symbol Type:

0 - Vericode  
1 - DataMatrix

Bits Per Cell:

2, 4, 8, or 16

Sample Width:

1 ... 8

Pre-Filtering:

0 - off  
1 - on

3 <Match String: string 199>

## **Vericode Sensor Query Response**

< Base Position Reference Sensor Data >

< Shape Information >

<Is Matrix Fixed : INT16> <Fixed Matrix Size : INT16>  
<Index Contrast : INT16> <Is Contrast Normal : INT16>  
<Edac Level : INT16> <Compression : INT16> <Symbol Type : INT16>  
<Bits Per Cell : INT16> <Sample Width : INT16> <Noise : INT16> <Pre-  
Filtering : INT16> <Filter Size : INT16> <Filter Iterations : INT16>  
<Use Packet: INT16> <Display Extracted Points: INT16>  
<Display Object Map : INT16> <Display Borders : INT16>  
<Display Vericode : INT16>  
  
<Match String: string 199>

Outputs:

<Inspection Result : INT32> <Output String : string 200>

| <b>Vericode Sensor Referencing IDs</b>          |     |
|-------------------------------------------------|-----|
| Input (i.e. – Will reference other sensors by): |     |
| Position                                        | 0   |
| Output (i.e. – Can be referenced by)            |     |
| Output String                                   | 128 |

| <b>Failure Code</b> | <b>Warning Code</b> | <b>Vericode SoftSensors</b> |
|---------------------|---------------------|-----------------------------|
| -1                  | 1                   | Failed Read                 |
| -2                  | 2                   | Failed Match String         |

## SmartLink SoftSensors (Module ID: 15)

### Command Format For Setting SmartLink Sensor Parameters

**SP** <sensor ID : INT16><parameter ID : UINT8><data>

<parameter ID> :

0 <Shape Type: UINT8>

NOTE: Value is not used

4 <Table ID Value: UINT8>

5 <Number of Columns: INT16> <Number of Rows: INT16>

NOTE: If new value for Columns or Rows is smaller than the previous value the data in the differing cells will be lost.

6 Cell Configuration Data

<Column Index: INT16> <Row Index: INT16> <Fail Character (ASCII value): UINT8> <Placeholder: INT8> <Configuration String Length: UINT16> <Configuration String: string>

Placeholder:

0 - Spaces

1 - Zeros

### SmartLink Sensor Query Response

<Coded String Length : UINT16> <Coded String : string>

<Table ID Value : UINT8>

<Filter Type: UINT8> <Filter Size: UINT8> <Filter Clipping: UINT16>

<Number of Sensors Referenced: INT16> <<Referenced Sensor ID: INT16>>

...

<Number of Columns: INT16> <Number of Rows: INT16>

<<Row Index: INT16> <Column Index: INT16> <Fail Character: UINT8>

<Placeholder: INT8> <Configuration String Length: UINT16>

<Configuration String: string>>

.  
.  
.

Outputs:

<Output String : string>

| <b>SmartLink Sensor Referencing IDs</b>                                                 |     |
|-----------------------------------------------------------------------------------------|-----|
| Input (i.e. – Will reference other sensors by):                                         |     |
| The SmartLink sensor will reference other sensors by any parameter DataLink can access. |     |
| Output (i.e. – Can be referenced by)                                                    |     |
| Output String                                                                           | 128 |

| <b>Failure Code</b> | <b>Warning Code</b> | <b>SmartLink SoftSensors</b>            |
|---------------------|---------------------|-----------------------------------------|
|                     |                     |                                         |
|                     |                     | Only standard sensor Failure Codes used |

## Snowflake SoftSensors (Module ID: 16)

### Command Format For Setting Snowflake Sensor Parameters

**SP** <sensor ID : INT16><parameter ID : UINT8><data>

<parameter ID> :

- 0 Set sensor shape/type
- 2 <Dot Size: UINT8> <Dot Tolerance: UINT8> <Squareability: UINT8>  
<Symmetry: UINT8> <Size Check: UINT8> <Quiet Zone Check: UINT8>  
<Neighbor Check: UINT8> <Perimeter Check: UINT8> <Density: UINT8>  
<Format: UINT8> <ECC: UINT8> <Color: UINT8> <Grid Alignment:  
UINT8> <Cell Sampling: UINT8>

Dot Size: 1 - 31  
Dot Tolerance: 1 - 31  
Size Check: 0, 1  
QuietZone Check: 0, 1  
Neighbor Check: 0, 1  
Perimeter Check: 0, 1  
Density: 0, 1-5, or 8-24  
Format: 0-5 or 255  
ECC: 0-4 or 255  
Color: 0, 1  
Grid Alignment: 0, 1  
Cell Sampling: 0 - 5

- 3 <Marking Flags: UINT32>  
Marking Flags (Bit Field):
  - 0 - Mark Intensity
  - 1 - Mark Search Grid
  - 2 - Mark Corner Check
  - 3 - Mark Grid Alignment
  - 4 - Mark Cell Sampling
  - 5 - Mark Corner Dots
  - 6 - Mark Left/Right Blobs
  - 7 - Mark Top/Bottom Blobs

- 9 <Check Match String: UINT8> <Match String: string 200>

### Snowflake Sensor Query Response

< Base Position Sensor Data >

< Shape Information >

<Dot Size: UINT8> <Dot Tolerance: UINT8> <Squareability: UINT8>  
<Symmetry: UINT8> <Size Check: UINT8> <Quiet Zone Check: UINT8>  
<Neighbor Check: UINT8> <Perimeter Check: UINT8> <Density: UINT8>  
<Format: UINT8> <ECC: UINT8> <Color: UINT8> <Grid Alignment: UINT8>  
<Cell Sampling: UINT8>

<Marking Flags: UINT32>

<Snowflake Library Version: string>

<Check Match String: UINT8> <Match String: string 200>

Outputs:

<Inspection Result : INT32> <Output String : string 200>

| Snowflake Sensor Referencing IDs                |     |
|-------------------------------------------------|-----|
| Input (i.e. – Will reference other sensors by): |     |
| Position                                        | 0   |
| Minimum Intensity                               | 16  |
| Maximum Intensity                               | 17  |
| Mean Intensity                                  | 18  |
| Median Intensity                                | 19  |
| Contrast                                        | 20  |
| Threshold                                       | 21  |
| Output (i.e. – Can be referenced by)            |     |
| Output String                                   | 128 |

| Failure Code | Warning Code | Snowflake SoftSensors |
|--------------|--------------|-----------------------|
| -1           | 1            | Failed Read           |
| -2           | 2            | Failed Match String   |

## Color SoftSensors (Module ID: 17)

### Command Format For Setting Pixel Counting and Color Monitoring Sensor Parameters

**SP** <sensor ID : INT16><parameter ID : UINT8><data>

<parameter ID> :

0 Set sensor shape/type

1 <Sensor Type: UINT8>

0 Color Monitoring

1 Pixel Counting

2 <Sampling Density X: INT16> <Sampling Density Y:INT16>

8 Warn Values

<Data 1: INT32> <Data 2: INT32> <Data3: INT32><Data4 :INT32>

9 Pass Values

<Data 1: INT32> <Data 2: INT32> <Data3: INT32><Data4 :INT32>

<Match Color Flag : INT8> < Digital Reselect Flag : INT8> <Match Color Name : String>

10 Settings:

<Sensor Type: INT8> <Color Space: INT8> <Sigma Factor: INT16>

<Marking Enabled Flag: INT8> <Matte Line Flag: INT8> <Matte Line Percentage Low: INT8> <Matte Line Percentage High: INT8>

<Highlight Line Flag: INT8> <Highlight Line Percentage: INT8>

<Type of Color Test: INT8> <Max Cluster Count: INT16><Min cluster Size: INT16> <Digital Relearn Flag: INT8> <Use Std Dev Flag:

UINT8> <Std Dev to Distance Ratio: INT8><Disable Sensor Flag: INT8>< Multi Learn Min Threshold: INT32><Digital Relearn Action: INT8>

42 <Color number to Overwrite base 1: INT16>

43 Test clustering

44 Learn Clustering

45 Learn a new Color

46 Merge colors

<Number of colors to merge: INT16>

-1 would merge all colors; any other number must be followed by the color indexes to merge of type INT16.

47 Set Color name

<Color Number base 1: INT32> <new name: STRING>

48 Delete color number base 1

<Color Number base 1: INT32>

-1 would delete all colors.

50     <Multi Learn Min Threshold: INT32><Multi Learn Action: INT8>  
           Multi Learn Action:  
           0- Delete All colors and Multi ReLearn  
           1- Multi Add  
           2- Multi Remove

54     <Color Sensor ID: INT16> <Color Number: INT32>  
       This command deletes all colors and then copies the specified  
       color number from specified color sensor ID into this color  
       sensor.

66     Used in Color monitoring; Update color data of the closest  
       matched color.

67     Update pixel count; the list of color indexes should follow same  
       as next case.

68     Show Colors:  
       <Number of colors to follow: INT8> followed by color indexes base  
       0 <First Color: INT8> ....

**SQ** <sensor ID : INT16> 0  
       ParamID of 0 would return the list of learned colors:  
       <Color number base 1: INT32> <Color Name: STRING> \n .... Next  
       color

### Examples:

#SP <sid> 42 2        Overwrite color number 2. It learns a color and updates the color  
 information of the color number 2. No new colors will be appended to the list.

#SP <sid> 43 test clustering.

#SP <sid> 44 learns all clusters. You can also use it in Color Monitoring

#SP <sid> 45 Learns a new color and appends it to the list

#SP <sid> 46 -1       merges all colors in the color sensor

#SP <sid> 47 2 newname   changes the name of color number 2

#SP <sid> 48 deletes all colors

#SP <sid> 48 2       deletes color number 2

#SQ <sid> 0 gets the list of learned colors

## **Color Sensor Query Response**

< Base Position Sensor Data >

< Shape Information >

<X Sampling Density: INT16> < Y Sampling Density: INT16>

<Warn 1: INT32><Warn 2: INT32><Warn 3: INT32><Warn 4: INT32><Fail 1: INT32><Fail 2: INT32><Fail 3:INT32><Fail 4: INT32>< Match Color Flag: INT8><Digital Reselect Flag: INT8>< Match Color Name: String>

<Pixel Counting Flag: INT8><Color Space: INT8><Sigma Factor: INT16><Marking Enabled: INT8><Matte Enabled: INT8><Matte Low Percentage: INT8><Matte High Percentage: INT8><High Light Enabled: INT8><High Light Percentage:INT8><Type Of Color Test: INT8><Max Cluster Count: INT16><Min Cluster Size: INT16><Digital Relearn Flag: INT8><Use Std. Dev.: UINT8><Std. Dev. Percentage: INT8><Disable Sensor Flag: UINT8><Multi Threshold: INT32><Digital Relearn Action: INT8><Number of Learned Colors:INT16>

Data for all learned colors.....

## **Outputs:**

<Delta Intensity:INT32><Delta Data 1:INT32><Delta Data 2: INT32><Delta Data 3:INT32><Pixel Count Percentage:INT32><Pixel Count:INT32><Match Index: INT8><Match Pixel Count:INT32>< Live Color Data>

## Segmentation SoftSensor (Module ID: 18)

### Command Format For Setting Segmentation Sensor Parameters

**SP** <sensor ID : INT16><parameter ID : UINT8><data>

<parameter ID> :

0 Set sensor shape/type

8 Sensor settings, ...

9 Pass and Warn Values

<Min # of segments Pass: INT32> <Max # of segments Pass: INT32>

<Max # of un-matched segments Pass: INT32><Min # of segments

Warn: INT32> <Max # of segments Warn: INT32> <Max # of un-matched segments Warn: INT32>

45 Learn all segments

54 <Color Sensor ID: INT16> <Color Number: INT32>

This command copies the specified color number from specified color sensor ID into background color of segmentation sensor.

67 Auto select reference segment

70 <Segment number to return results of (base 0): INT32>

### Segmentation Sensor Query Response

< Base Position Sensor Data >

< Shape Information >

<Rule Type:INT8><Abs. Register Design:INT8><Color Var.

Enabled:INT8><Edge Detection Enabled:INT8><Color Variance: INT16><Edge Detection Threshold:INT16><Min Segment Size:INT16><Marking

Enabled:INT8><Seed Type:INT8><Min number of Segments Pass:INT32><Max number of Segments Pass: INT32><Max Unmatched segments Pass:INT32><Min

number of Segments Warn:INT32><Max number of Segments Warn: INT32><Max Unmatched segments Warn:INT32><Has Background Color:INT8><Color

Data><Segmentation Matching Rules><Number of Learned

Segments:INT16><Segment Data for all learned segments><Matching Rules for all learned segments>

### Outputs:

<Total Segment Count:INT32><Matched Segment Count:INT32><Failed Segment Count:INT32><X Position: INT32><Y Position: INT32><Theta 0

Position:INT32><Theta 1 Position: INT32><Delta X:INT32><Delta

Y:INT32><Delta Color:INT32><Delta Pixel Count Percentage:INT32>

## Spectrograph SoftSensor (Module ID: 19)

### Command Format For Setting Spectrograph Sensor Parameters

**SP** <sensor ID : INT16><parameter ID : UINT8><data>

<parameter ID> :

- 1 <Use Std Dev: UINT8><Std Ratio: INT8><Find Peak: UINT8><FOV Balance: UINT8><Peak Normalization: UINT8><White Reference: UINT8>< White Reference Sensor ID: INT16><Graph Data Type: UINT8><Process Data Type: UINT8><Window: UINT32><Abs Color: UINT8>
- 2 <Subtract Black Image: UINT8><Spectral Bin Count: UINT8><Spatial Bin Count: UINT8>
- 8 <Warn 1:INT32><Warn 2: INT32><Warn 3: INT32><not used: INT32>
- 9 <Pass 1: INT32><Pass 2: INT32><Pass 3: INT32><not used: INT32><Match Spectrum: INT8><Digital reselect: INT8><String name: String>
- 10 <not used: INT8><Digital Relearn: UINT8>
- 42 <Spectrum Index:INT16>  
Overwrite
- 45 Learn a new Spectrum.
- 47 <Spectrum number to rename base 1: INT32><New name: String>
- 48 <Spectrum number to delete (base 1, -1 removes all): INT32>

**SQ** <sensor ID : INT16> 0

ParamID of 0 would return the list of learned Spectrums:

<Spectrum number base 1: INT32> <Spectrum Name: STRING> \n ...  
Next Spectrum

### Spectrograph Sensor Query Response

< Base Position Sensor Data >

< Shape Information >

<Warn 1: INT32><Warn 2: INT32><Warn 3: INT32><Warn 4: INT32><Fail 1: INT32><Fail 2: INT32><Fail 3:INT32><Fail 4: INT32>< Match Spectrum Flag: INT8><Digital Reselect Flag: INT8>< Match Spectrum Name: String>

<Marking Enabled: INT8><Digital Relearn Flag: INT8><Use Std. Dev.: UINT8><Std. Dev. Percentage: INT8><Find Peak: UINT8><Do FOV Balance:UINT8><Normalize To Peak:UINT8><Has White Reference:UINT8><Processing Data Type:UINT8><Do Black Subtraction:UINT8><Spectrum Bin Count: UINT8><Spatial Bin

*Count:UINT8><Norm Window:UINT32><Do Abs. Color:UINT8><Number of Learned Spectrums:INT16>*

Data for all learned Spectrums.....

**Outputs:**

*<Starting Wave Length:INT32 ><Ending Wave Length:INT32><Peak Intensity:INT32><Peak Wave Length:INT32><X Value:INT32><Y Value:INT32><Z Value:INT32><L :INT32><a\* : INT32><b\* :INT32><c\*:INT32><H: INT32><Delta Peak Intensity:INT32><Delta Peak Location:INT32><Delta 3: INT32><Match Index: INT8><Match Delta:INT32><Match Spectrum Name: String>*