

---

# Universal Serial Bus OEM Point-of-Sale Device Interface Specification

Document Number (USBOEMINTERFACEV2.3.DOC)

Version Code:(Ver.2.3)  
Owners: Susan Brosnan  
Kumar Kori  
Mark Ring

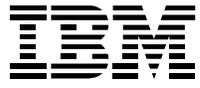
Approved by Jeff Harmon  
Mike Hyde  
on  
10/07/2011  
Address questions  
to:  
Techline

(800)426-9990

<http://www2.clearlake.ibm.com/store/support/html/plquestion.html>

## **DISCLAIMER**

IBM assumes no liability for any use of the information contained herein. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of IBM or third parties. THE INFORMATION CONTAINED IN THIS DOCUMENT IS PROVIDED ON AN 'AS IS' BASIS. Therefore, IBM reserves the right to make changes to its products, other product information, and this publication without prior notice. NO WARRANTIES OF ANY KIND, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE ARE OFFERED IN THIS DOCUMENT.



# Document Control Information

## Location

Repository for this document

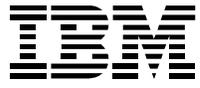
### CURRENCY STATEMENT

This version of the document was printed from an on-line system and must only be used for reference purposes. The official copy of this document is the on-line version. Please preserve the integrity of the documentation by destroying any obsolete versions and by not removing any pages from this printed copy.

## Revision History

Changes resulting in document revisions will be summarized in this table in chronological sequence. Revision bars (|) will highlight the text changed in new document versions.

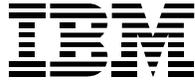
Version	Approval Date	Change Description
V1.0	04/01/98	Initial release of the document (no change bars).
V1.1	04/27/98	Add interface descriptors, flash in progress status bit, pseudo RS232 device command set,
V1.2	06/19/98	Incorporate feed back from Mobinetix, Symbol Technologies. Merged Hand-held and Table-top Scanner specifications.
V1.21	06/23/98	Remove reenumeration circuitry.
V1.22	08/06/98	Incorporate internal and PSC feed back.
V1.23	08/26/98	Incorporate feed back from Hobart.
V1.24	02/23/99	Add Powered USB specifications.
V1.25	10/22/99	Updated USB hardware information and Scanner/Scale interfaces.
V1.26	04/05/00	Corrections and clarifications.
V1.27	05/16/00	Direct I/O Response clarifications
V1.28	02/05/01	Changes in Power Requirements and Flash Update specification; clarifications in the Scanner and Scale chapters
V1.29	10/31/02	Changes to Pseudo-RS232 Interface Additional label type identifiers for Reduced Space Symbology (RSS), coupons Report descriptor clarifications Scale command response clarifications
V2.0	01/26/04	Add label type definitions for bar codes supported in JavaPOS V1.7 but not in V1.29 of this specification Add configuration bits/commands for additional bar code types Add Scanner II interface specification chapter.
V2.1	10/12/2006	Modify Scanner II specification
V2.2	09/20/2010	Add Label ID definitions for new bar codes since V2.1 Remove Scanner II specification
V2.3	10/07/11	Add Symbologies to 3.3.1 Enable/Disable Additional Symbologies Correct section reference number errors in Section 4



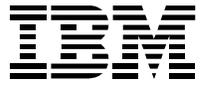
<b>DOCUMENT CONTROL INFORMATION.....</b>	<b>3</b>
<b>1 OVERVIEW.....</b>	<b>12</b>
1.1 INTRODUCTION.....	12
1.2 RELEVANT PUBLICATIONS.....	12
<b>2 COMMON REQUIREMENTS FOR ALL DEVICES.....</b>	<b>12</b>
2.1 MECHANICAL REQUIREMENTS.....	12
2.1.1 Powered USB Connector (Optional).....	12
2.2 ELECTRICAL REQUIREMENTS.....	13
2.2.1 Device Enumeration Without Hot Plugging (Mandatory).....	13
2.3 POWER REQUIREMENTS.....	13
2.3.1 5-volt specification (Mandatory).....	13
2.3.2 12-volt specification (Mandatory).....	13
2.3.3 24-volt specification (provided for information only).....	13
2.4 FIRMWARE REQUIREMENTS.....	14
2.4.1 Human Interface Device (Mandatory).....	14
2.4.2 String Descriptor (Mandatory).....	14
2.4.3 Firmware Flash Update (Optional).....	15
2.4.3.1 Interface Descriptor.....	15
2.4.3.2 Report Descriptor.....	15
2.4.3.3 Firmware Update Operation.....	15
2.4.3.4 Commands (Output Report).....	16
2.4.3.5 File Format for Firmware Update.....	16
2.4.3.6 Status (Input Report).....	19
2.4.3.7 Error Recovery.....	19
2.4.3.8 Requirements for Coexisting Interfaces.....	19
2.4.3.9 Topology Considerations.....	19
<b>3 SCANNER.....</b>	<b>19</b>
3.1 INTRODUCTION.....	19
3.2 COMMAND/RESPONSE SET.....	20
3.2.1 Command Format.....	20
3.2.2 Command Description.....	20
3.2.2.1 Enable Scanner.....	20
3.2.2.2 Disable Scanner.....	20
3.2.2.3 Enable Beeper.....	21
3.2.2.4 Disable Beeper.....	21
3.2.2.5 Configure Scanner.....	21
3.2.2.6 Report Scanner Configuration.....	27
3.2.2.7 Configure EAN/JAN-13 Two-Label Flags.....	27
3.2.2.8 Report EAN/JAN-13 Two-Label Flag Configuration.....	28
3.2.2.9 Direct I/O Command.....	28
3.2.2.10 Test Request.....	28
3.2.2.11 Status Request.....	29

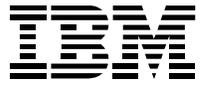
3.2.2.12 System Reset.....	29
3.2.3 <i>Response Format</i> .....	29
3.2.4 <i>Response Description</i> .....	30
3.2.4.1 Command Reject.....	30
3.2.4.2 Enable Scanner.....	31
3.2.4.3 Disable Scanner.....	31
3.2.4.4 Enable Beeper.....	31
3.2.4.5 Disable Beeper.....	31
3.2.4.6 Successful Configuration.....	31
3.2.4.7 Coerced Configuration.....	31
3.2.4.8 Unsuccessful Configuration.....	32
3.2.4.9 Configuration Data.....	32
3.2.4.10 Successful EAN/JAN-13 Two-Label Flag Configuration.....	32
3.2.4.11 Unsuccessful EAN/JAN-13 Two-Label Flag Configuration.....	32
3.2.4.12 EAN/JAN-13 Two-Label Flag Configuration Data.....	32
3.2.4.13 Successful Direct I/O Command.....	33
3.2.4.14 Unsuccessful Direct I/O Command (Unaccepted).....	33
3.2.4.15 Successful Direct I/O Command Which Returns Data.....	33
3.2.4.16 Unsuccessful Direct I/O Command (Not Allowed).....	33
3.2.4.17 Unsuccessful Direct I/O Command (Not Defined).....	33
3.2.4.18 Label Data.....	34
3.2.4.19 Status Request.....	35
3.2.4.20 Test Request.....	35
3.3 DIRECT I/O COMMANDS.....	35
3.3.1 <i>Enable/Disable Additional Symbologies</i> .....	36
3.3.2 <i>Device Information</i> .....	37
3.4 FIRMWARE CONSIDERATIONS.....	37
3.4.1 <i>Descriptors</i> .....	38
3.4.1.1 Interface String Descriptor.....	38
3.4.1.2 Report Descriptor.....	38
<b>4 SCALE.....</b>	<b>39</b>
4.1 INTRODUCTION.....	39
4.2 COMMAND/RESPONSE SET.....	39
4.2.1 <i>Command Format</i> .....	39
4.2.2 <i>Command Description</i> .....	39
4.2.2.1 English Weight Request.....	39
4.2.2.2 Metric Weight Request.....	40
4.2.2.3 Zero Scale.....	40
4.2.2.4 Enable Extended (3-Byte) Status.....	40
4.2.2.5 Disable Extended (3-Byte) Status.....	40
4.2.2.6 Configure Scale.....	41
4.2.2.7 Report Scale Configuration.....	42
4.2.2.8 Clear Remote Display.....	42
4.2.2.9 Test Request.....	43

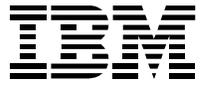
4.2.2.10 Status Request.....	43
4.2.2.11 System Reset.....	43
4.2.3 <i>Response Format</i> .....	43
4.2.4 <i>Response Description</i> .....	44
4.2.4.1 Command Reject.....	44
4.2.4.2 Successful Configuration.....	45
4.2.4.3 Unsuccessful Configuration.....	45
4.2.4.4 Configuration Data.....	45
4.2.4.5 Unacceptable Command.....	45
4.2.4.6 English Weight Request (Four-digit weight).....	45
4.2.4.7 English Weight Request (Five-digit weight).....	46
4.2.4.8 Metric Weight Request.....	46
4.2.4.9 Successful Zero Scale.....	46
4.2.4.10 Unsuccessful Zero Scale.....	46
4.2.4.11 Enable Extended Status.....	46
4.2.4.12 Disable Extended Status.....	47
4.2.4.13 Successful Clear Remote Display.....	47
4.2.4.14 Unsuccessful Clear Remote Display.....	47
.....	47
4.2.4.15 Status Request.....	47
4.2.4.16 System Reset.....	47
4.2.4.17 Test Request.....	47
4.3 FIRMWARE CONSIDERATION.....	48
4.3.1 <i>Descriptors</i> .....	48
4.3.1.1 Interface String Descriptor.....	48
4.3.1.2 Report Descriptor.....	48
<b>5 PSEUDO-RS232 INTERFACE.....</b>	<b>49</b>
5.1 INTRODUCTION.....	49
5.2 COMMAND/RESPONSE SET.....	49
5.2.1 <i>Command Format</i> .....	49
5.2.2 <i>Command Description</i> .....	50
5.2.2.1 Setup.....	50
5.2.2.2 Channel Control.....	50
5.2.2.3 Read Status.....	51
5.2.2.4 Transmit.....	51
5.2.2.5 Receive Buffer.....	51
5.2.2.6 Test Request.....	51
5.2.2.7 Status Request.....	52
5.2.2.8 System Reset.....	52
5.2.3 <i>Response Format</i> .....	52
5.2.4 <i>Response Description</i> .....	53
5.2.4.1 Command Reject.....	53
5.2.4.2 Setup.....	53
5.2.4.3 Channel Control.....	53



5.2.4.4 Read Status/Status Request.....	53
5.2.4.5 Transmit.....	53
5.2.4.6 Transmit Complete.....	54
5.2.4.7 Device Data.....	54
5.2.4.8 Receive Buffer.....	54
5.2.4.9 Receive Buffer (Data Buffer Error).....	54
5.2.4.10 Reset.....	55
5.2.4.11 Test Request.....	55
5.3 FIRMWARE CONSIDERATION.....	55
5.3.1 <i>Descriptors</i> .....	56
5.3.1.1 Interface String Descriptor.....	56
5.3.1.2 Report Descriptor.....	56







# 1 Overview

## 1.1 Introduction

This document describes a means for OEM devices to connect seamlessly to the IBM USB POS network. Once they conform to this specification, OEM devices will have the software support from IBM for the following environments:

- ✓ IBM POSS for Windows (Microsoft Windows XP, Microsoft Windows 7, Microsoft WEPOS, Microsoft POSReady)
- ✓ OPOS (Microsoft Windows XP, Microsoft Windows 7, Microsoft WEPOS, Microsoft POSReady)
- ✓ JavaPOS (Microsoft Windows XP, Microsoft Windows 7, Microsoft WEPOS, Microsoft POSReady, 4690 OS, Linux)
- ✓ 4690 Operating System

In order for IBM to provide the software and device driver support mentioned above, mandatory mechanical, hardware, and firmware requirements for the devices will be spelled out in detail later in this document.

## 1.2 Relevant Publications

- ❖ IBM Point of Sale Programming Reference and User's Guide.
- ✓ IBM Point of Sale Installation, Keyboards, and Code Pages.
- ✓ Universal Serial Bus Specification, Revision 1.1, September 23, 1998.
- ✓ Universal Serial Bus Device Class Definition for Human Interface Devices (HID), Firmware Specification, Version 1.0 Final.

# 2 Common Requirements For All Devices

This section describes requirements which are applicable to all devices. Requirements which are specific to the device will be discussed in the appropriate device sections.

## 2.1 Mechanical Requirements

### 2.1.1 Powered USB Connector (Optional)

---

IBM POS system units will have Powered USB connectors available for devices which require more than 500mA of current or voltage greater than 5 volts. In addition, the Powered USB connector provides better mechanical retention as compared to standard USB connectors. Pinouts of assigned voltages are documented in EIA standard number EIA-700BAAD. 12 volts is assigned to key position 1 and 24 volts is assigned to key position 2.

IBM POS system units will have multiple 12 volt connectors but only one 24 volt connector which is normally used for POS printers.

## 2.2 Electrical Requirements

### 2.2.1 Device Enumeration Without Hot Plugging (Mandatory)

This capability is required if the device firmware update is needed or when a reset command is sent down to the device. This can be accomplished by electrically removing the USB connection sense resistor from the circuit.

## 2.3 Power Requirements

### 2.3.1 5-volt specification (Mandatory)

The USB POS devices shall comply to the USB specifications v.1.1.

### 2.3.2 12-volt specification (Mandatory)

Voltage/Tolerance (at system unit)	Steady-state (normal operating): 12 V +/- 10 % Transient: the device must tolerate 12 V +10% / -20% for up to 500 us, to support hotplugging of other devices on the same system. This is expected to occur infrequently. See also "In-rush/Hotplug Device Current Limitation" below .
Individual Device Current per Port	1.5 A - average and peak current - except during in-rush/hotplug
Total I/O Current	4.0 A - average and peak current - except during in-rush/hotplug
In-Rush/Hotplug Current Limitation	The device shall present no more than a 12 mA-sec load with a peak current not to exceed 20A (equivalent to charging a perfect 1000uF capacitor through a 600 milliohm resistor).

### 2.3.3 24-volt specification (provided for information only)

Voltage (at system unit)	24 V +2.4 v, -0.4v
Individual Current per Port	2.0 A RMS
In-Rush Current	Current should be limited to 2.0A during in-rush; adherence to this specification will allow hot-plugging

## 2.4 Firmware Requirements

### 2.4.1 Human Interface Device (Mandatory)

All POS USB devices supported by this specification shall conform to the "USB Device Class Definition for Human Interface Device (HID) Firmware Specification--4/7/99 Version 1.1". All Devices shall be USB high speed devices.

### 2.4.2 String Descriptor (Mandatory)

According to the USB specification, the Serial Number, Manufacturer, Product ID, and Interface string descriptors are optional. However, these strings are mandatory in the IBM USB POS network. These strings are used by IBM software to uniquely identify individual devices in a network where multiple identical devices exist.

Following are examples of three different string descriptors for the following strings: IBM © 1998, POS - NVRAM, FLASH, S/N 00001 Rev. 00.01. Refer to string descriptor section (Chapter 9) in the USB spec for more detail.

Field	Offset/Size (Bytes)	Description	Value
<i>bLength</i>	0/1	Length of String descriptor, in bytes	0x04
<i>bDescriptorType</i>	1/1	Descriptor type = STRING	0x03
<i>bString</i>	2/2	Array of LangID codes (in this case, the 2-byte code for English)	0x0409
<i>bLength</i>	4/1	Length of String descriptor, in bytes	0x16
<i>bDescriptorType</i>	5/1	Descriptor type = STRING	0x03
<i>bString</i>	6/20	Manufacturer	IBM © 1998
<i>bLength</i>	26/1	Length of String descriptor, in bytes	0x26
<i>bDescriptorType</i>	27/1	Descriptor type = STRING	0x03
<i>bString</i>	28/36	Product - POS Hub (NVRAM & Cash Drawers)	POS - NVRAM, FLASH
<i>bLength</i>	64/1	Length of String descriptor, in bytes	0x2A
<i>bDescriptorType</i>	65/1	Descriptor type = STRING	0x03
<i>bString</i>	66/40	Product - Serial Number s)	S/N 00001 Rev. 00.01

## 2.4.3 Firmware Flash Update (Optional)

---

Depending on the operating system, the firmware flash update software support may vary. To take advantage of this infrastructure, the device firmware needs to follow certain guidelines. Once the device receives the data, it is the device responsibility to check for data integrity as well as burn the data into its flash memory.

The device firmware flash update mechanism shall have its own interface/endpoint. The firmware flash update shall use the Vendor Defined Page 1 (FF45h). The following usages are defined for the firmware flash update purposes :

ÿA000h: Flash Report

ÿA001h: Flash Commands and associated data

ÿA002h: Flash Status

### 2.4.3.1 Interface Descriptor

The flash interface string descriptor shall be as follows:

**Flash Update (Usage = A000h, Usage Page = FF45h)**

### 2.4.3.2 Report Descriptor

Usage Page (Vendor Defined Page 1)

Usage (Flash Report)

Collection (Application)

Report Size (8)

Report Count (263)

Logical Minimum (0)

Logical Maximum (255)

Usage (Flash Command)

Output (Data, Var, Abs)

Report size (8)

Report count (8)

Usage (Flash Status)

Input (Data, Var, Abs)

End Collection

### 2.4.3.3 Firmware Update Operation

The flash update operation will be initiated from the system unit when it sends a Start command. Upon receiving the last record (identified by the zero data count), the device will force reenumeration (by reset if necessary in the particular device hardware) and switch to the new code load to respond to the descriptor requests and configuration command from the system unit. From that point on the code will always detect that there is a complete new code load to be used on each power on (check of permanent record kept in the flash).

### 2.4.3.4 Commands (Output Report)

Byte Number	Name	Value or Bit Location ( 7 MSB, 0 LSB)	Meaning
0, 1	Reserved	N/A	Undefined -- Do Not Use
2	Command Byte # 1		
		00h	Command Byte # 2 = System Command
		01h	Start Flash Update Operation
		02h	Write Record
3	Command Byte # 2		
		10h	Test Command
		20h	Status Request
		40h	Reset Device and Reenumerate
4	Data Count (DC)		Data Count of Flash Data Notes: 1) Does not include 3-byte fixed field 2) Data count = 0 ⇒ last record to be written
5, 6, 7	Fixed Field		Not included in Data Count (DC); can be used for EEPROM address, if applicable
0	Flash Data		New firmware to be written, 255 bytes or less

Notes:

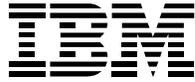
1) Minimum Output Report length is four. Determination of the actual length should be based on Command Byte #1. If the command is one that requires data, use Data Count to determine the length; if the command does not require data, the length is four.

2) Microsoft Windows operating systems always pad the Output Report and send 263 bytes.

### 2.4.3.5 File Format for Firmware Update

The firmware developer is responsible for creating the download file for the device drivers. In order for IBM software to initiate the flash update of a device, the firmware update file must be called XXXYYYY.DAT where XXX is the three-letter identifier of the OEM vendor and YYYY is the USB Product ID of the device.

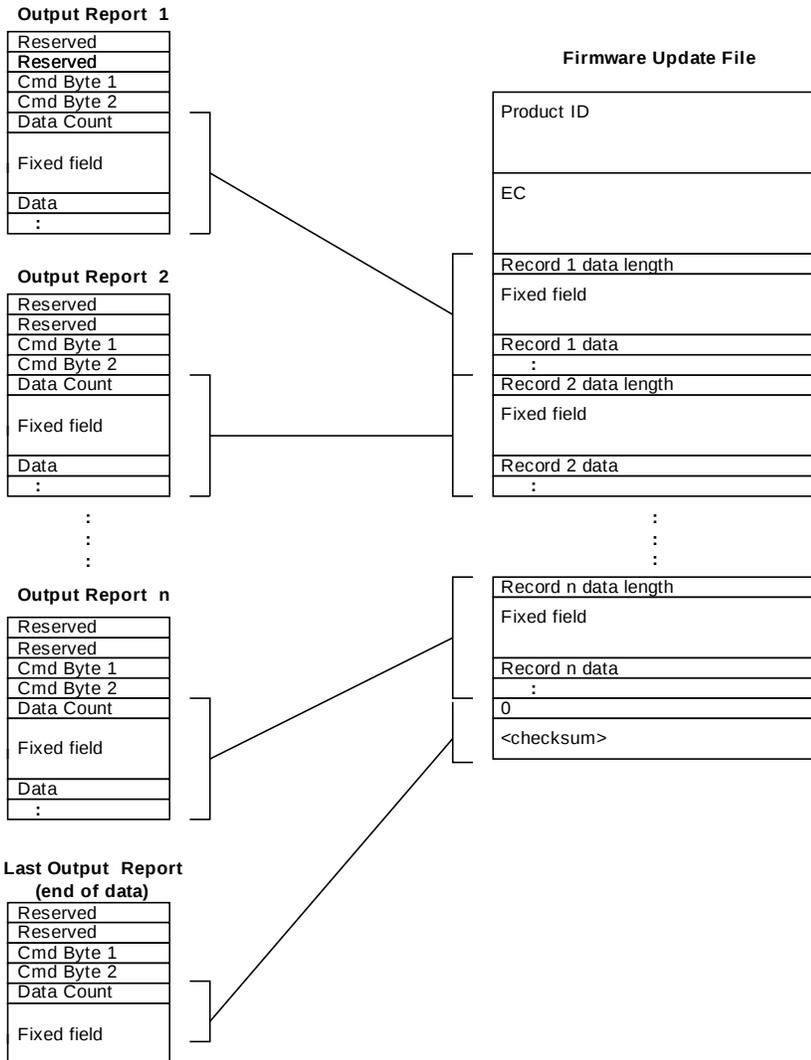
The following file format is supported:



Size	Sub-structure	Description
4 bytes		4-byte Product ID in ASCII, representation of hexadecimal Product ID normally sent in USB descriptors
4 bytes		4-byte device EC in ASCII, regular format -- i.e. V.100='0010'
Record 1 (variable length)	1 byte	Length of Record 1 Data (in binary, does not include length of "Fixed field")
	3 bytes	Fixed field
	xx bytes	Record 1 Data (in binary)
Record 2 (variable length)	1byte	Length of Record 2 Data (in binary, does not include length of "Fixed field")
	3 bytes	Fixed field
	xx bytes	Record 2 Data (in binary)
:	:	:
:	:	:
:	:	:
Record n (variable length)	1 byte	Length of Record n Data (in binary, does not include length of "Fixed field")
	3 bytes	Fixed field
	xx bytes	Record n Data (in binary)
1 byte		End of operation = 0x00
2 bytes		Checksum; format is device dependent
Size	Sub-structure	Description
4 bytes		4-byte Product ID in ASCII, representation of hexadecimal Product ID normally sent in USB descriptors
4 bytes		4-byte device EC in ASCII, regular format -- i.e. V.100='0010'
Record 1 (variable length)	1 byte	Length of Record 1 Data (in binary, does not include length of "Fixed field")
	3 bytes	Fixed field
	xx bytes	Record 1 Data (in binary)
Record 2 (variable length)	1byte	Length of Record 2 Data (in binary, does not include length of "Fixed field")
	3 bytes	Fixed field
	xx bytes	Record 2 Data (in binary)

For each record in the Firmware Update File, an Output Report will be generated using the length of record data, Fixed field, and record data fields from the Firmware Update File to fill in the Data Count, Fixed field, and Flash Data fields in the Output Report. See the diagram on the next page.

### Output Reports generated from Firmware Update File



### 2.4.3.6 Status (Input Report)

Byte Number	Name	Value or Bit Location (7 MSB, 0 LSB)	Meaning
0	Status Byte # 1		
		Bit 7	Command Reject
		Bit 6	Operation in Error
		Bit 5-2	Reserved for future use, do not assume a value.
		Bit 1	Flash Load in Progress
		Bit 0	Command Complete (does not imply no errors ⇒ see other bits for additional information)
1	Status Byte # 2		Reserved for future use, do not assume a value.

### 2.4.3.7 Error Recovery

Error recovery of the flash operation is to be handled by the flash device driver and the device firmware. When in doubt the device driver must restart the operation by sending the start command at any time. However, once a flash operation is started it must be completed before other device operations can be resumed.

If the device driver receives the error status from the device saying that the operation did not complete correctly it should restart the flash operation and attempt it again. Upon a second failure the device should be serviced.

### 2.4.3.8 Requirements for Coexisting Interfaces

All device interfaces must define a status bit in their input reports to tell the system unit when the device is not available due to flash upgrade of the device being in progress. The location of the status bit or value is device specific.

### 2.4.3.9 Topology Considerations

The IBM SurePOS family of system units might not have any external root ports. All ports can be via a USB hub. This should be taken into consideration in firmware design and debug.

## 3 Scanner

### 3.1 Introduction

The information provided in this section applies to any USB scanner. The interface described is based on the IBM 4698 Scanner interface. Two of these scanners can be attached per register/lane -- see 3.4 *Firmware Considerations* which contains information concerning the usages defined for scanner devices.

## 3.2 Command/Response Set

### 3.2.1 Command Format

The scanner command is fixed length (11 bytes) and is laid out as follows:

1st	2nd	3rd-11th
Command Byte 1	Command Byte 2	Data Bytes

where:

Command byte 1	Command byte 2	Reader command
00h	10h	Test
00h	20h	Status request
00h	40h	Reset
11h	00h	Scanner enable
12h	00h	Scanner disable
14h	00h	Beeper enable
18h	00h	Beeper disable
20h	00h	Configure scanner
21h	00h	Report scanner configuration
23h	00h	Configure EAN/JAN-13 Two-Label Flags
34h	00h	Report EAN/JAN-13 Two-Label Flag configuration
30h	xxh	Direct I/O command

Data bytes are used for the Configure Scanner, Configure EAN/JAN-13 Two-Label Flags, and Direct I/O commands. They are zero (00h) otherwise.

### 3.2.2 Command Description

#### 3.2.2.1 Enable Scanner

Command: **11h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h**

This command enables the scanner to read bar codes.

Scanner response: Send three-byte status indicating “scanner enabled”. See section 3.2.4 *Response Description* for details.

#### 3.2.2.2 Disable Scanner

Command: **12h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h**

This command disables the scanner.

Scanner response: Send three-byte status indicating “scanner not enabled”. See section 3.2.4 *Response Description* for details.

### 3.2.2.3 Enable Beeper

Command: **14h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h**

This command enables the scanner beeper. The Enable Beeper command overrides the default beeper state set by the “Configure Scanner” command.

Scanner response: Send three-byte status indicating “good-read beep enabled”. See section 3.2.4 *Response Description* for details.

### 3.2.2.4 Disable Beeper

Command: **18h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h**

This command disables the scanner beeper. The Disable Beeper command overrides the default beeper state set by the “Configure Scanner” command.

Scanner response: Send three-byte status indicating “good-read beep not enabled”. See section 3.2.4 *Response Description* for details.

### 3.2.2.5 Configure Scanner

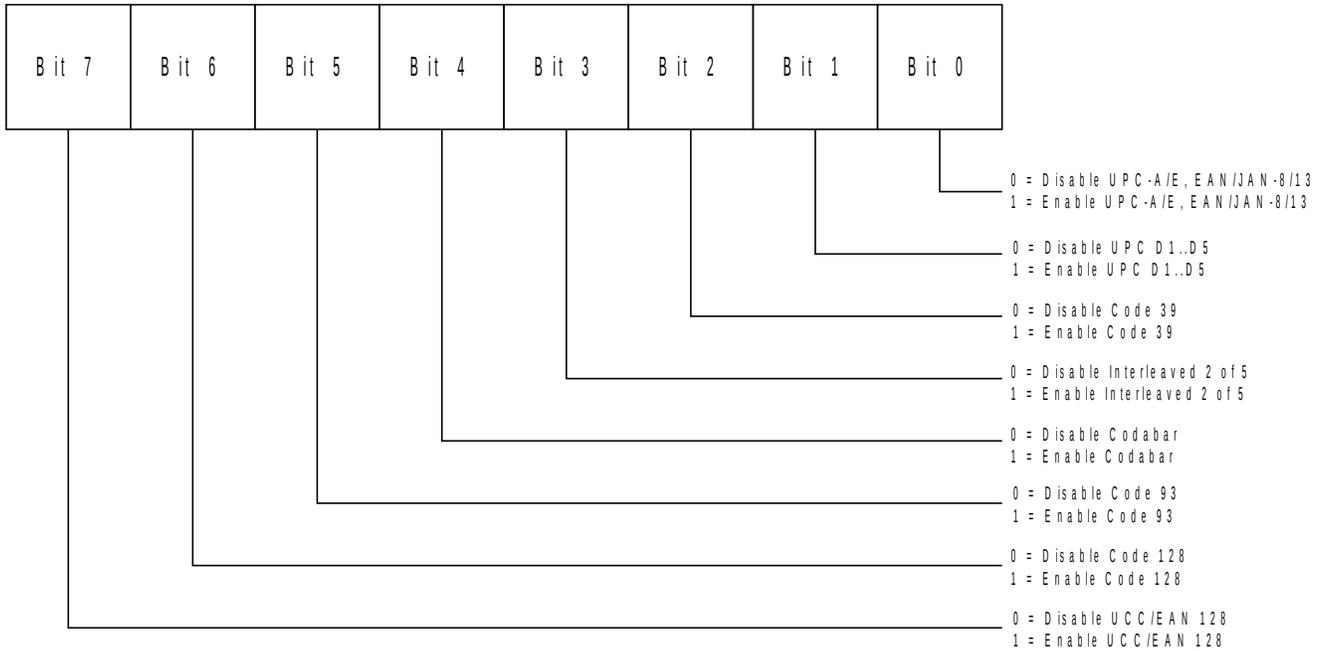
Command: **20h, 00h, cfg0, cfg1, cfg2, cfg3, cfg4, cfg5, cfg6, cfg7, cfg8**

This command specifies the desired configuration for the scanner. The nine (9) data bytes in this command represent a set of scanner capabilities typical of retail point-of-sale scanners. Scanner operating parameters not specified by this command can be configured via the Direct I/O command (section 4.2.2.9).

Scanner response: Send three-byte status indicating “configuration successful”, “configuration coerced”, or “configuration not successful”. A scanner device can conform to this specification without supporting all of the capabilities represented as long as the device responds appropriately when an unsupported capability is requested. See section 3.2.4 *Response Description* for details.

The configuration data bytes are described on the following pages.

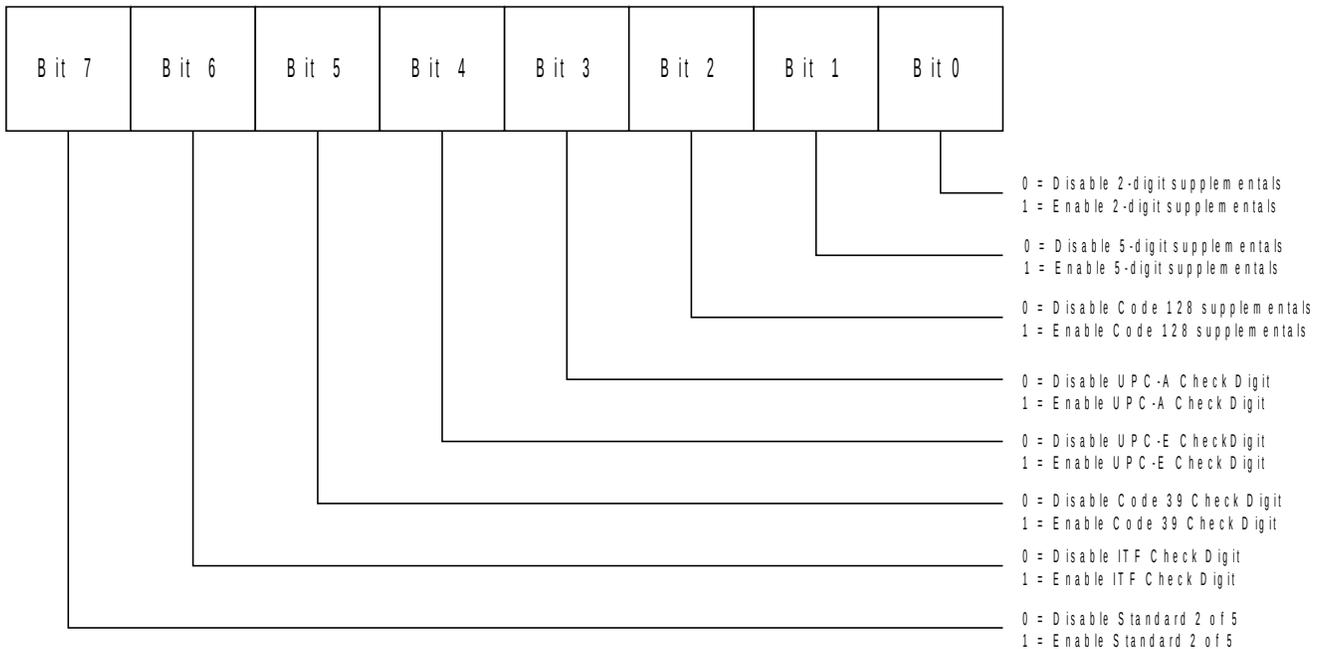
cfg0:



**NOTES:**

1. 0x00 is not a valid value for this configuration byte.
2. Enable/Disable Code 128 (bit 6) has no affect on Code 128 supplementals.

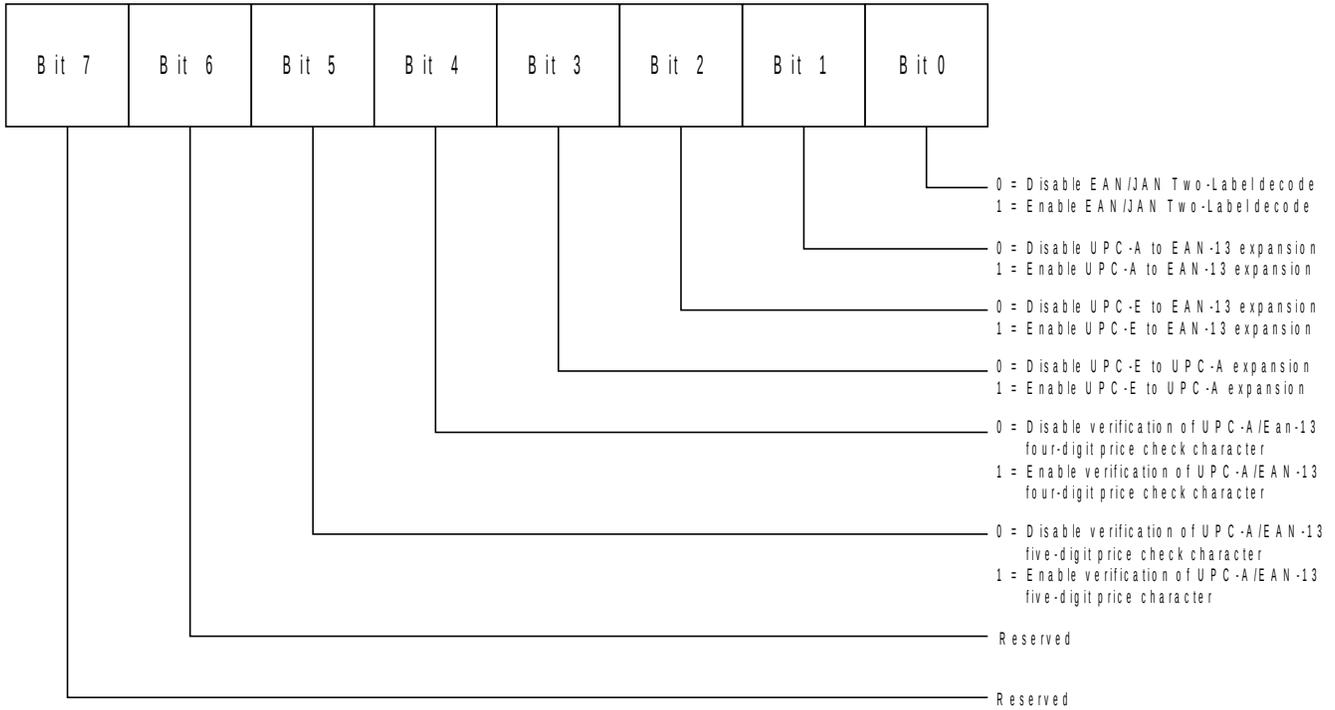
cfg1:



**NOTES:**

1. When enabled, supplementals are **optional** for all selected symbologies, recognized but not required.
2. When disabled, check digits are not transmitted to the POS terminal but they are still verified by the scanner.

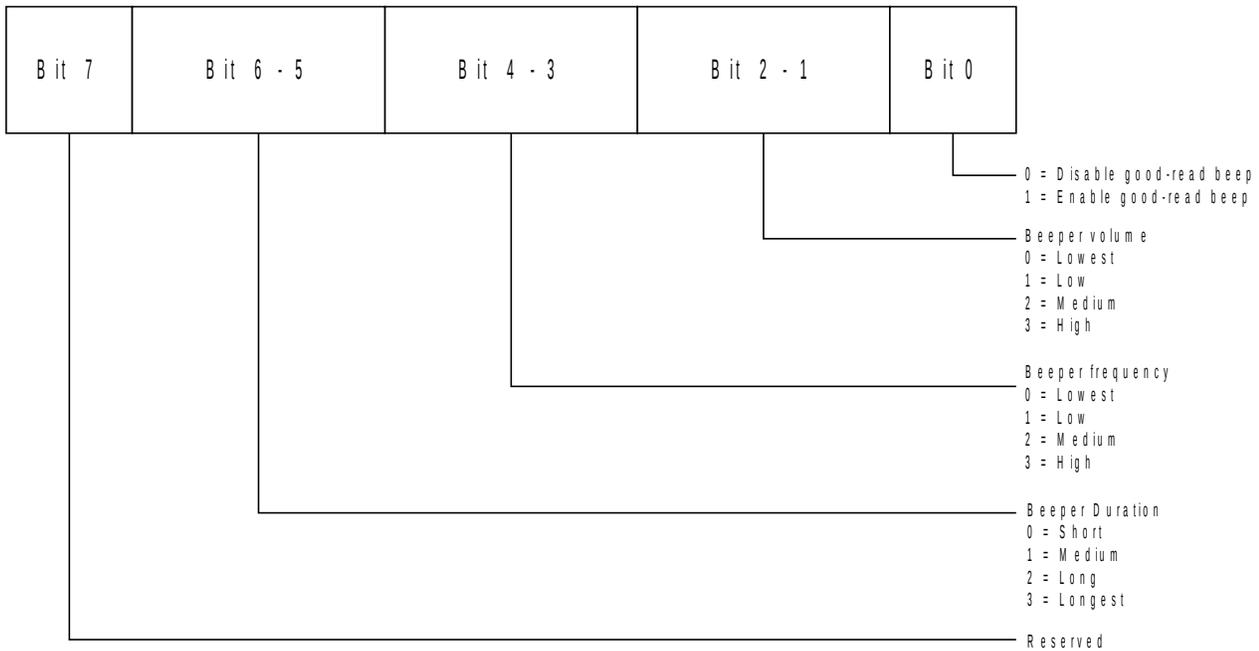
cfg 2:



**NOTES:**

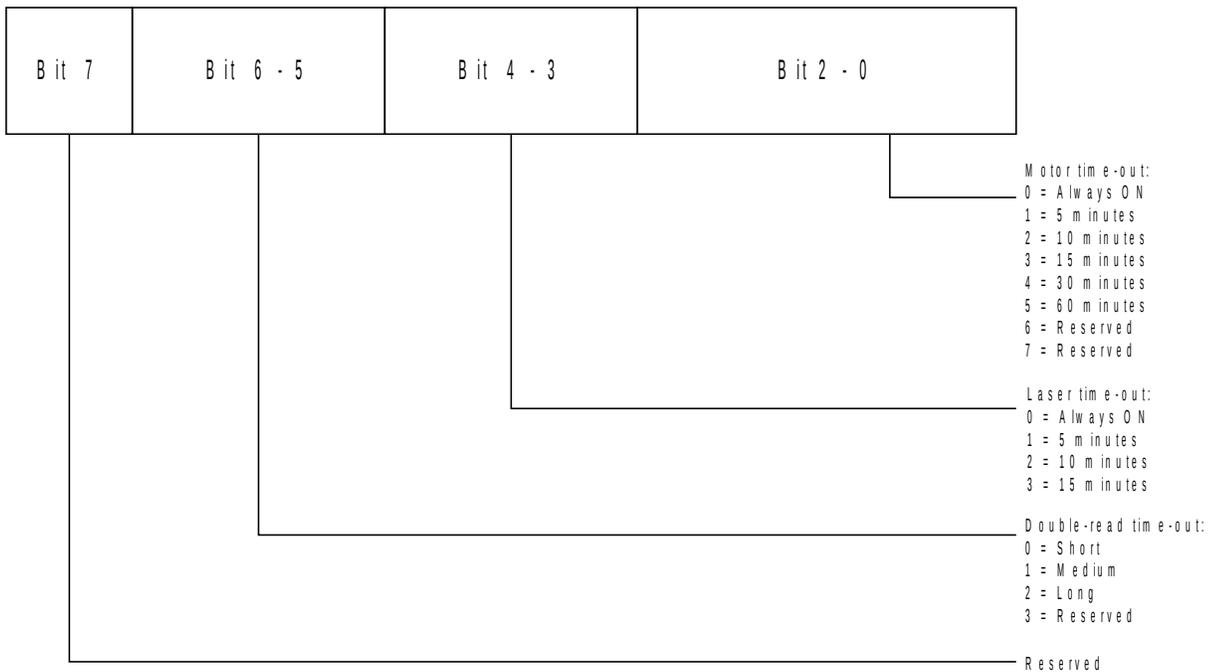
1. Bits 2 and 3 are mutually exclusive.
2. Bits 4 and 5 are mutually exclusive.

cfg 3:

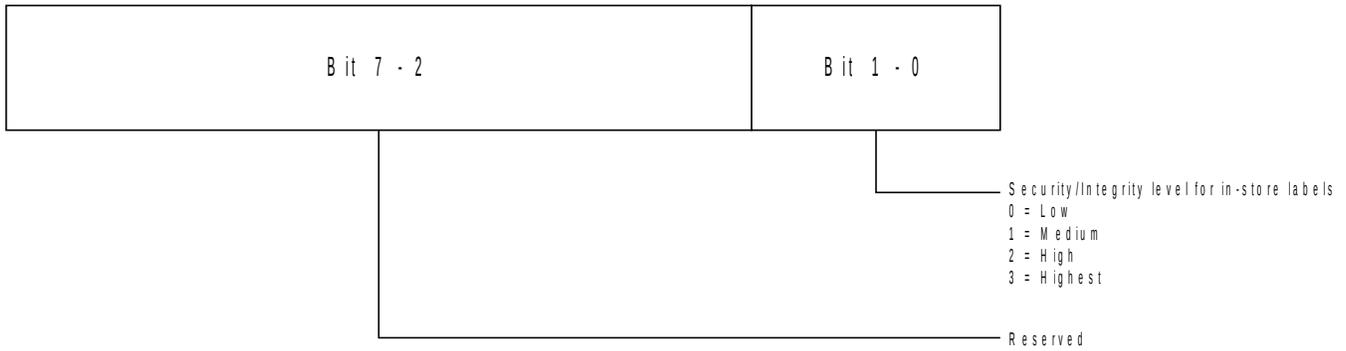


**NOTE:** Bit 0 will indicate the default state of the beeper following a reset. This can be overridden using the "Enable Beeper" and "Disable Beeper" commands.

cfg 4:



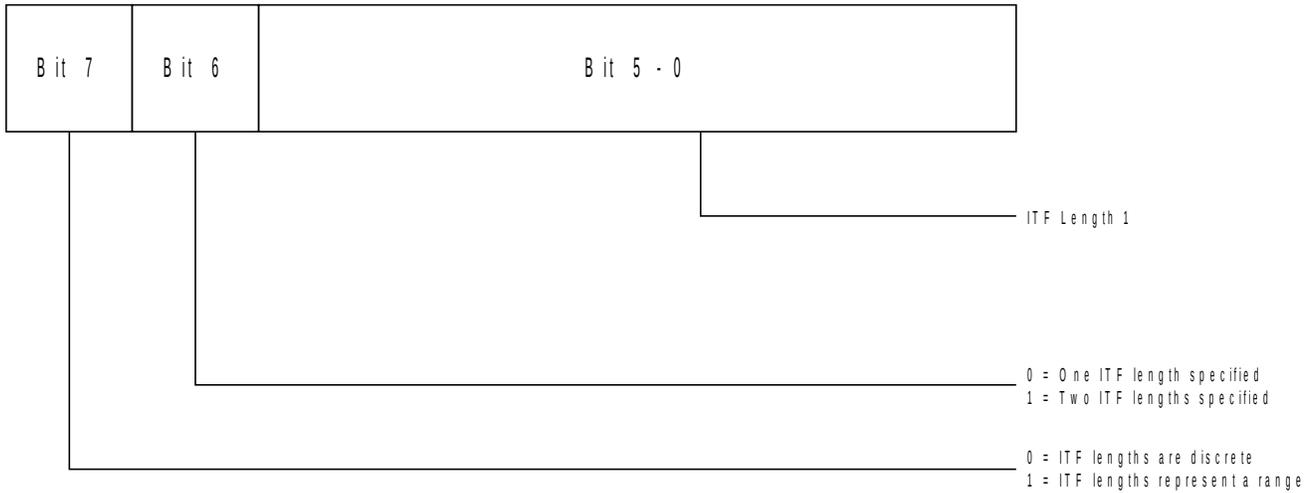
cfg 5:



**NOTE:**

In-store labels: UPCA with number system character 2 and 4 and EAN-8/13 with flag one equal to 2.

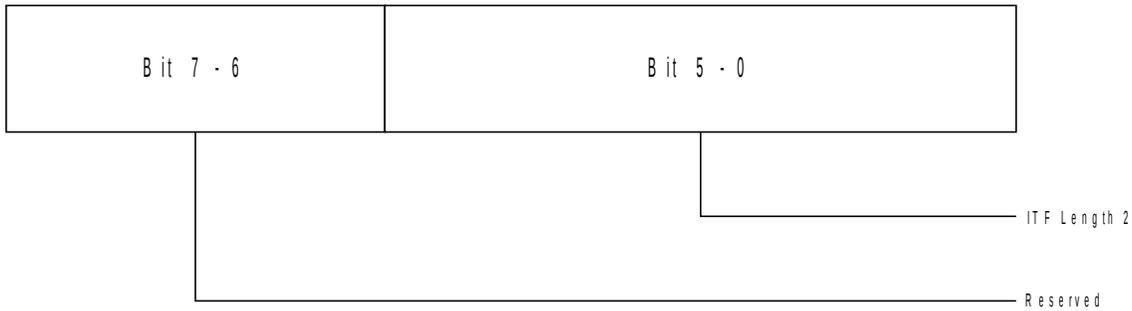
cfg 6:



**NOTES:**

1. This configuration byte is ignored unless the Interleaved 2 of 5 (ITF) symbology is enabled.
2. If the ITF symbology is enabled, bits 0-5 must contain an even value between 4 and 32 (inclusive).

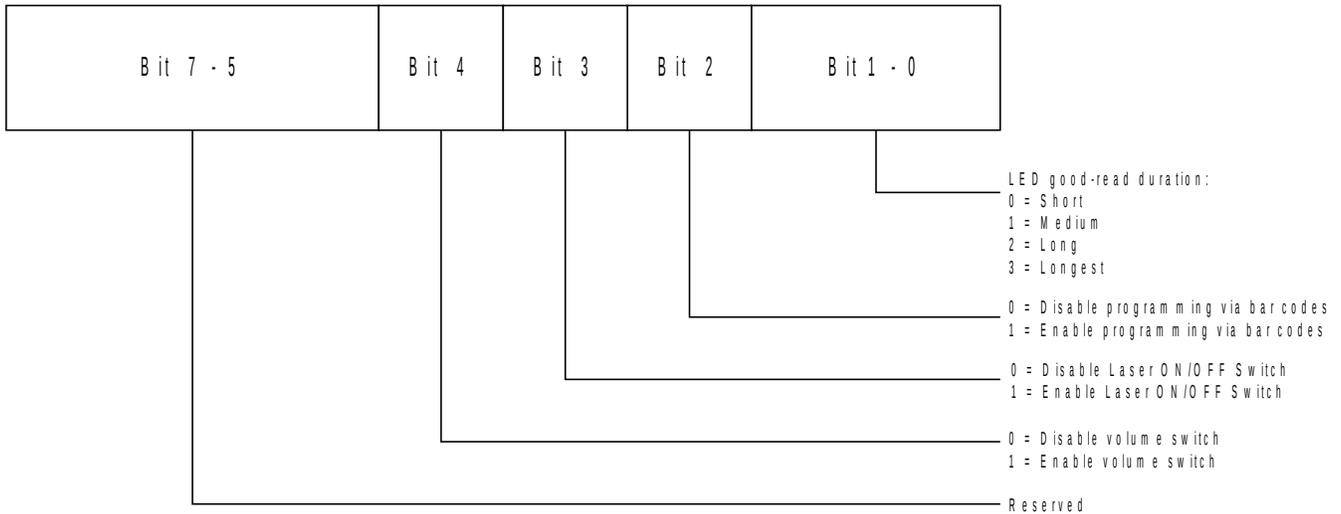
cfg 7:



**NOTES:**

1. Bits 0-5 are ignored unless the Interleaved 2 of 5 (ITF) symbology is enabled.
2. If the ITF symbology is enabled, bits 0-5 must be 0 unless two lengths are being configured (cfg 6, bit 6 set).
3. If two ITF lengths are being configured, bits 0-5 must contain an even value between 4 and 32 (inclusive).

cfg 8:





Flag values are specified in Binary Coded Decimal (BCD) format in the decimal range 00-99. None of the 1st-label flags shall match any of the 2nd-label flags; there can be duplicate 1st-label flags, duplicate 2nd-label flags, and duplicate flag pairs. Four flag pairs (eight bytes) **must** be included with this command.

Scanner response: Send three-byte status indicating “EAN/JAN-13 Two-Label configuration successful” or “EAN/JAN-13 Two-Label configuration not successful”. A scanner device can conform to this specification without supporting this command as long as the device responds appropriately. See section 3.2.4 *Response Description* for details.

### 3.2.2.8 Report EAN/JAN-13 Two-Label Flag Configuration

Command: **34h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h**

This command returns the scanner EAN/JAN-13 two-label flag configuration bytes as described in section 3.2.2.7 *Configure EAN/JAN-13 Two-Label Flags*. Eight (8) data bytes are returned.

Scanner response: Send three-byte status indicating “EAN/JAN-13 two-label flag configuration data frame” plus scanner EAN/JAN-13 two-label flag configuration data (8 bytes). A scanner device can conform to this specification without supporting this command as long as the device responds appropriately. See section 3.2.4 *Response Description* for details.

### 3.2.2.9 Direct I/O Command

Command: **30h, dio0, dio1, xxh, xxh, xxh, xxh, xxh, xxh, xxh**

This command is intended to allow the host access to device-specific commands, particularly commands which affect scanner configuration parameters that are not addressed by the Configure Scanner command. The device driver will not interpret the data passed to the device on a Direct I/O command.

Scanner response: Send three-byte status indicating “Direct I/O command response frame” and “Direct I/O command accepted”. A scanner device can conform to this specification without supporting this command as long as the device responds appropriately. See section 3.2.4 *Response Description* for details.

See section 3.3 *Direct I/O Commands* for some commands that have been defined by IBM.

### 3.2.2.10 Test Request

Command: **00h, 10h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h**

This command requests that the scanner perform its self-test.

Scanner response: Send three-byte status. See section 3.2.4 *Response Description* for details.

### 3.2.2.11 Status Request

Command: **00h, 20h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h**

This command requests status from the scanner.

Scanner response: Send three byte status. See section 3.2.4 *Response Description* for details.

### 3.2.2.12 System Reset

Command: **00h, 40h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h**

This command resets the scanner.

Scanner response: Refer to section 3.2.4 *Response Description* for scanner response.

## 3.2.3 Response Format

1st	2nd	3rd	4th	5th	6th - Nth
Response Length	Status byte 0	Status Byte 1	Status Byte 2	Continuation Byte or Data Byte	Data Bytes

where:

✓ Response length is the total length of the response including length byte, status bytes, and data bytes. The maximum length for a single response frame is 64 bytes.

✓ Status Byte 0 is defined as follows (bit 0 is the least significant bit):

Bit Position	Meaning
0	Flash in progress (If Flash update is implemented)
1	Configuration data response frame
2	EAN/JAN-13 Two-Label configuration data response frame
3	Reserved - must be 0
4	0=Good-read beep disabled; 1=Good-read beep enabled
5	Hardware error
6	Direct I/O command data response frame
7	Device not ready to receive commands

NOTE: "Device not ready to receive commands" should be returned in response to any supported command that cannot be processed by the scanner microcode at the time it is received.

V Status Byte 1 is defined as follows (bit 0 is the least significant bit):

Bit Position	Meaning
0	Scanner Alive (powered on)
1	0=Scanner disabled; 1=Scanner enabled
2	Data length/check digit error
3	Reserved - always 0
4	Reserved - always 0
5	Reserved - always 0
6	Reserved - always 0
7	Undefined command received (command reject)

V Status Byte 2 is defined as follows (bit 0 is the least significant bit):

Bit Position	Meaning
0	Configuration successful
1	Configuration coerced -- See section 3.2.4.6
2	EAN/JAN-13 Two-Label Flag configuration successful
3	Direct I/O command accepted
4	Direct I/O command not allowed
5	Direct I/O command undefined
6	Reserved - always 0
7	Reserved - always 0

Y The Continuation Byte only applies to Direct I/O command responses. A value of 00h indicates that this is the last Direct I/O response frame. For all other commands, the 5th response byte is a Data Byte.

## 3.2.4 Response Description

---

The scanner firmware shall pad the response to 64 bytes. Refer to section 3.3.1.1 *Report Descriptor* for details.

### 3.2.4.1 Command Reject

Response: **04h, 00xx0000b, 10000x1b, 00000000b**

This response is returned after the scanner receives an undefined or unsupported command.

#### 3.2.4.2 Enable Scanner

Response: **04h, 00xx0000b, 00000011b, 00000000b**

This response is returned following the receipt of an Enable Scanner command.

#### 3.2.4.3 Disable Scanner

Response: **04h, 00xx0000b, 00000001b, 00000000b**

This response is returned following the receipt of a Disable Scanner command.

#### 3.2.4.4 Enable Beeper

Response: **04h, 00x10000b, 000000x1b, 00000000b**

This response is returned following the receipt of an Enable Beeper command.

#### 3.2.4.5 Disable Beeper

Response: **04h, 00x00000b, 000000x1b, 00000000b**

This response is returned following the receipt of an Disable Beeper command.

#### 3.2.4.6 Successful Configuration

Response: **04h, 00xx0000b, 000000x1b, 00000001b**

This response is returned following the successful execution of a Configure Scanner command. This response indicates that the scanner has accepted the configuration exactly as specified in the configuration command.

#### 3.2.4.7 Coerced Configuration

Response: **04h, 00xx0000b, 000000x1b, 00000011b**

This response is returned following the successful execution of a Configure Scanner command. This response indicates that the configuration has been accepted by the scanner but has been altered in some way. If the device does not support particular configuration features specified in the configuration command, it can accept the configuration anyway but only set bits in its internal configuration for the features it does support.

### 3.2.4.8 Unsuccessful Configuration

Response: **04h, 00xx0000b, 000000x1b, 00000000b**

This response is returned following the receipt of invalid parameter specifications in a Configure Scanner command. This response indicates that the scanner's configuration has not been modified.

### 3.2.4.9 Configuration Data

Response: **0Dh, 00xx0010b, 000000x1b, 00000000b, cfg0, cfg1, cfg2, cfg3, cfg4, cfg5, cfg6, cfg7, cfg8**

This response is returned following the receipt of the Report Configuration command. The three-byte status is returned followed by nine (9) data bytes. The response to this command should contain the actual scanner configuration which may not match the configuration requested in a previous Configure Scanner command. See the section 3.2.2.5 *Configure Scanner* for the format of cfg0..cfg8.

### 3.2.4.10 Successful EAN/JAN-13 Two-Label Flag Configuration

Response: **04h, 00xx0000b, 000000x1b, 00000100b**

This response is returned following the successful execution of a Configure EAN/JAN-13 Two-Label Flags command.

### 3.2.4.11 Unsuccessful EAN/JAN-13 Two-Label Flag Configuration

Response: **04h, 00xx0000b, 000000x1b, 00000000b**

This response is returned following a Configure EAN/JAN-13 Two-Label Flags command in which any of the first label flag values match any of the second label flag values or an incorrect number of data bytes was included. If the EAN/JAN-13 Two-Label Flag Configuration command itself is not supported, the device response should be Command Reject (see section 3.2.4.1).

### 3.2.4.12 EAN/JAN-13 Two-Label Flag Configuration Data

Response: **0Ch, 00xx0100b, 000000x1b, 00000000b, flag pair 1, flag pair 2, flag pair 3, flag pair 4**

This response is returned following the receipt of a Report EAN/JAN-13 Two-Label Flag Configuration command. The three-byte status is returned followed by eight (8) data bytes (each flag pair is two bytes). See the section 3.2.2.7 *Configure EAN/JAN-13 Two-Label Flags* for the format of flag pair 1..flag pair 4.

If the Report EAN/JAN-13 Two-Label Flag Configuration command is not supported, the device response should be Command Reject (see section 3.2.4.1).

### 3.2.4.13 Successful Direct I/O Command

Response: **04h, 00xx0000b, 000000x1b, 00001000b**

This response is returned following the successful execution of a Direct I/O command. When no data is returned by the device in response to a Direct I/O command, Status Byte 0, Bit 6 must not be set.

### 3.2.4.14 Unsuccessful Direct I/O Command (Unaccepted)

Response: **04h, 00xx0000b, 000000x1b, 00000000b**

This response is returned following the unsuccessful execution of a Direct I/O command. That is, the Direct I/O command is supported by the device, the data portion of the Direct I/O command contains a valid Direct I/O subcommand, but some other portion of the data is unacceptable to the device.

If the Direct I/O command itself is not supported, the device response should be Command Reject (see section 3.2.4.1).

### 3.2.4.15 Successful Direct I/O Command Which Returns Data

Response: **length byte, 01xx0000b, 000000x1b, 00001000b, cont, xxh, ... xxh**

This response is returned following the successful execution of a Direct I/O command. When data is returned by the device in response to a Direct I/O command, Status Byte 0, Bit 6 and Status Byte 2, Bit 3 must be set.

### 3.2.4.16 Unsuccessful Direct I/O Command (Not Allowed)

Response: **04h, 00xx0000b, 000000x1b, 00010000b**

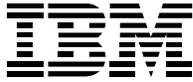
This response is returned following the receipt of a Direct I/O Command which is not allowed. That is, the Direct I/O command is supported by the device, the data portion of the Direct I/O command contains a valid Direct I/O subcommand, but the device is unable to process the specified subcommand at this time.

If the Direct I/O command itself is not supported, the device response should be Command Reject (see section 3.2.4.1).

### 3.2.4.17 Unsuccessful Direct I/O Command (Not Defined)

Response: **04h, 00xx0000b, 000000x1b, 00100000b**

This response is returned following the receipt of a Direct I/O Command which is not defined. That is, the Direct I/O command is supported by the device but the data portion of the Direct I/O command does not contain a valid Direct I/O subcommand. If the Direct I/O command itself is not supported, the device response should be Command Reject (see section 3.2.4.1).



### 3.2.4.18 Label Data

Response: **length byte, 00xx0000b,00000x11b, 00000000b, xxh, xxh, ..., xxh**

The first three bytes of this response are the three-byte status. The data bytes contain label data followed by one to three label type indicator bytes. The label data is transmitted in ASCII for all label types except UPC-A, UPC-E, EAN/JAN-8, EAN/JAN-13 and Discrete 2 of 5, which can be transmitted in ASCII or Binary Coded Decimal (BCD).

Label Type	Label Length	Label Type Identifier
EAN/JAN-13	13	16h
EAN/JAN-8	8	0Ch
UPC-A	12	0Dh
UPC-E	8	0Ah
UPC-D1	14	0011h
UPC-D2	20	0012h
UPC-D3	24	0014h
UPC-D4	28	0017h
UPC-D5	32	001Dh
UPC-A+2	14	00160Bh
UPC-A+5	17	00110Bh
UPC-E+2	10	00120Bh
UPC-E+5	13	00140Bh
EAN/JAN-8+2	10	00170Bh
EAN/JAN-8+5	13	001D0Bh
EAN/JAN-13+2	15	00130Bh
EAN/JAN-13+5	18	00150Bh
Standard (Discrete) 2 of 5	<= 32	000C0Bh
Interleaved 2 of 5	4 <= length <= 32	000D0Bh
Code39	<= 32	000A0Bh
Code128	<= 32	00180Bh
Codabar	< = 32	000E0Bh
Code 93	<= 32	00190Bh
UCC/EAN-128	<= 64	00250Bh
UPC-A w/Code 128 Supplemental	Base:12, Supplemental: <=64	00200Bh
UPC-E w/Code 128 Supplemental	Base: 8, Supplemental: <=64	00210Bh
EAN/JAN-8 w/Code 128 Supplemental	Base: 8, Supplemental: 4 - 30	00220Bh
EAN/JAN-13 w/Code 128 Supplemental	Base:13, Supplemental: 4 - 30	00230Bh
GS1 DataBar/GS1 DataBar Limited/GS1 DataBar Stacked	16	002A0Bh
GS1 DataBar Expanded	<= 42 (alpha), <=72 (numeric)	002B0Bh
PDF417	<=2710	002E0Bh
Maxicode*	<=93	002F0Bh

OCR-A*		00300Bh
OCR-B*		00310Bh
DataMatrix**	<= 2335 alpha-numeric	00320Bh
QR Code**	<= 7366 (numeric), <= 4464 (alpha) MicroQR Code <= 35 numeric	00330Bh
Aztec Code**	<= 3067 (alpha), <= 3832 (numeric)	00340Bh
Code 49**	<= 192	00350Bh
Unknown*	<= 64	00FF0Bh

Notes:

- 1) For label types other than EAN/JAN-13, EAN/JAN-8, UPC-A, and UPC-E, the first byte of the label identifier is a label data continuation character -- 00h indicates the last block of label data, 10h indicates that there is more label data to come.
- 2) UPC Coupons shall be transmitted as a single label with the appropriate label type identifier bytes.
- 3) Label types followed by an "\*" indicate labels that are not currently supported by IBM's 4690 Operating Systems and will be rejected as an invalid label type.
- 4) Label types followed by an "\*\*" indicate labels that are not currently supported by IBM POSS for Windows, IBM OPOS and IBM JavaPOS drivers or IBM's 4690 Operating System and will be rejected as an invalid label type.
- 5) Composite Labels of Type CC-A, CC-B and CC-C should be sent in as their composite parts. That is, the linear bar code portion should be sent in with its appropriate label ID and the composite portion should be sent in separately with its appropriate label ID.

### 3.2.4.19 Status Request

Response: **04h, x0xx0000b, 000000x1b, 00000000b**

This response is returned after receiving a Status Request command. System Reset  
There is no response for the Reset command. However, the scanner shall initialize, perform basic assurance test (BAT), and re-enumerate.

### 3.2.4.20 Test Request

Response: **04h, x0xx0000b, 000000x1b, 00000000b**

This response is returned after receipt of a Test Request command.

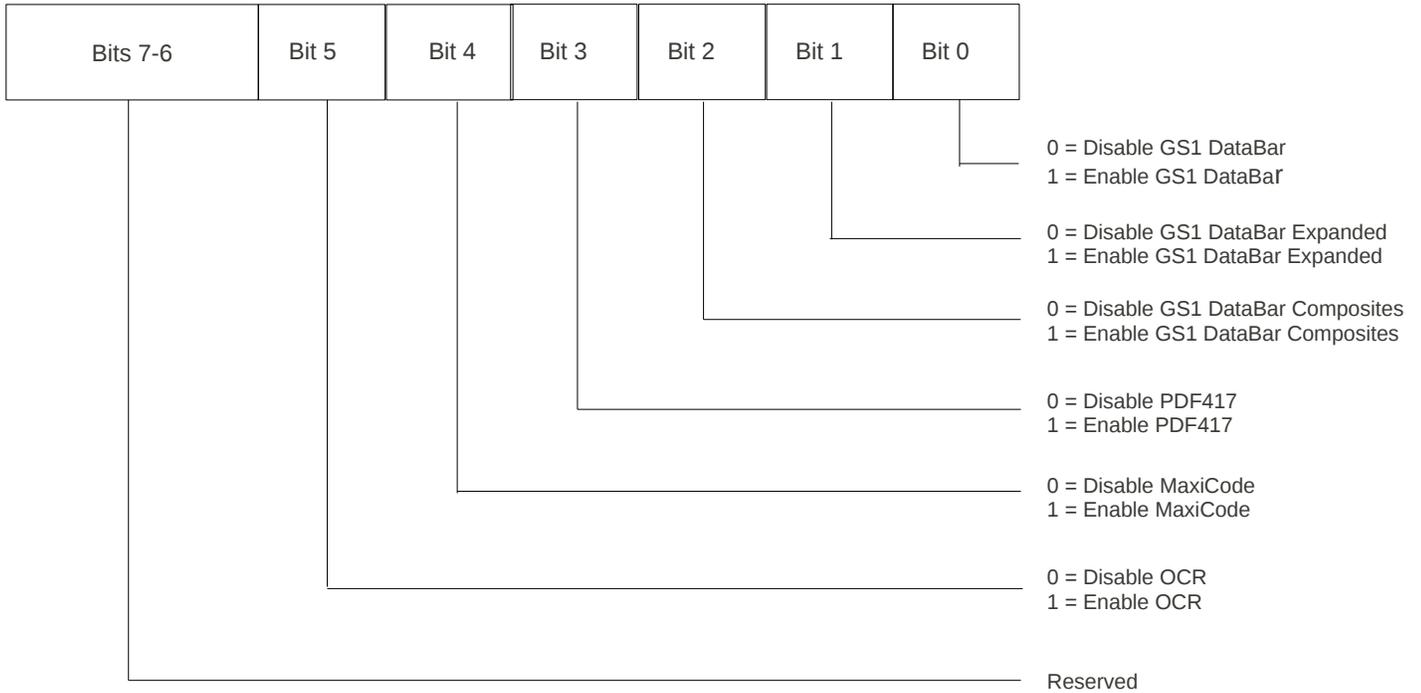
## 3.3 Direct I/O Commands

IBM reserves Direct I/O commands (cmd0, cmd1) in the range of F0h, 00h through FFh, FFh.

### 3.3.1 Enable/Disable Additional Symbologies

Command: **30h, FFh, FFh, sym0, sym1, sym2, sym3, sym4, sym5, sym6, sym7**

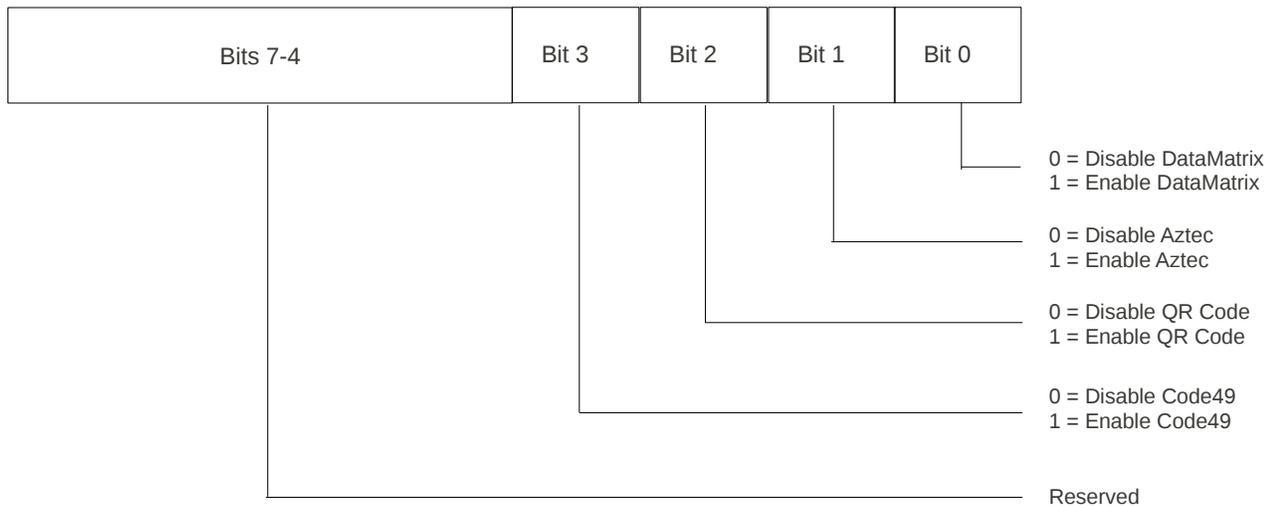
sym 0:



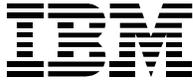
**NOTES:**

- GS1 DataBar (bit 0) includes GS1 DataBar, GS1 DataBar Stacked and GS1 DataBar Limited

sym 1:



sym2 – sym7 : undefined/reserved (must be 0)



sym2-sym7 : undefined/reserved (must be 0)

Response: 04h, 00xx0000b, 000000x1b, 0000x000b

This command does not return any data so Status Byte 0, Bit 6 is not set. If the command was successful, Status Byte 2, Bit 3 will be set.

---

### 3.3.2 Device Information

---

Command: 30h, FFh, FEh, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h

Returns to the caller information about the device including a vendor/manufacture identification.

Response: xxh, 01xx0000b, 000000x1b, 0000x000b, cont, VendorId1, VendorId2, vs0, vs1, ..., vsN

**VendorId1, VendorId2:** Two-byte vendor identifier in big-endian format (low order byte first). The identifier returned is the identifier assigned to the manufacturer by USB.ORG.

**vs0 - vsN:** Vendor-specific device information. It is assumed that all devices from the same manufacturer will return their device information in the same format.

## 3.4 Firmware Considerations

The scanner shall use a Vendor Defined Page 1 (FF45h) usage page for all of its usages. There are 4 usages defined for the scanner in this page:

- V Table-top Scanner: 4A00h
- V Hand-held Scanner: 4B00h
- V Scanner Command: 4A01h
- V Scanner Status: 4A02h

**The command and status usages are shared to all devices of this type.**

Due to a constraint in the Linux USB infrastructure (usbdevfs.o), early releases of the IBM Linux POS subsystem cannot guarantee the interrupt polling specified by the device (in the end point descriptor). In the laboratory environment, the observed polling periods are in the range from a few microseconds to hundreds of milliseconds depending on the system load. An OEM device vendor is expected to be able to handle this condition properly in the Linux environment. IBM is actively working with the Linux open source community to eliminate this constraint.

## 3.4.1 Descriptors

---

Device, configuration, interface, and endpoint descriptors are device specific.

### 3.4.1.1 Interface String Descriptor

The scanner interface string descriptors shall be as follows:

**Table-top Scanner (Usage = 4A00h, Usage Page = FF45h)**

**Hand-held Scanner (Usage = 4B00h, Usage Page = FF45h)**

The Interface Descriptor for the Scanner Interface should only use values of 0x00 for the Subclass and Protocol fields. Specifically, do not use 'Boot Keyboard' values of 0x01 for these fields as it will cause unwanted interaction between the IBM BIOS and the Scanner.

### 3.4.1.2 Report Descriptor

The scanner shall have one output and one input reports to send commands and receive status/data respectively. The output report will have a fixed length of 11 bytes. The input report will have a fixed length of 64 bytes. Under the HID device model, output reports will be sent out using a control write on end point 0. Input reports will be sent in on an endpoint other than zero (device specific).

For example, the report descriptor for the table-top scanner shall be as follows:

```
Usage Page (Vendor Defined Page 1)
Usage (Table-top Scanner)
Collection (Application)
  Usage (Table-top Scanner Command)
  Report Size (8)
  Report Count (11)
  Logical Minimum (0)
  Logical Maximum (255)
  Output (Data, Var, Abs)
  Usage (Table-top Scanner Status)
  Report Count (64)
  Input (Data, Var, Abs)
End Collection
```

## 4 Scale

### 4.1 Introduction

The information provided in this section applies only to scales. The interface described is based on the IBM 4698 Scale interface.

### 4.2 Command/Response Set

#### 4.2.1 Command Format

The reader command is fixed length (5 bytes) and is laid out as follows:

1st	2nd	3rd-5th
Command byte 1	Command byte 2	Data bytes 1-3

where:

Command byte 1	Command byte 2	Scale command
00h	10h	Test
00h	20h	Status request
00h	40h	Reset
01h	00h	English weight request
02h	00h	Metric weight request
03h	00h	Zero scale
04h	00h	Enable extended (3-byte) status
05h	00h	Disable extended (3-byte) status
06h	00h	Clear remote display
20h	00h	Configure scale
21h	00h	Report scale configuration

Data bytes are used only for the Configure Scale command. They are zeros (00s) otherwise.

#### 4.2.2 Command Description

##### 4.2.2.1 English Weight Request

Command: **01h, 00h, 00h, 00h, 00h**

This command requests weight data in pounds from the scale. Weight data is returned in Binary Coded Decimal (BCD) format with a resolution of either 1/100 or 1/1000 pounds. Weight data will not be included in the response if weight data is not available, the scale is not configured for English mode, or the scale is configured for a remote

display and no display is detected. If the scale is not configured for English mode, the “Unacceptable Command” status bit (status byte 0, bit 6) shall be set.

Scale response: Send two- or three-byte status and four- or five-digit weight data if available and allowed. See section 4.2.4 *Response Description* for details.

#### 4.2.2.2 Metric Weight Request

Command: **02h, 00h, 00h, 00h, 00h**

This command requests weight data in kilograms from the scale. Weight data is returned in BCD format with a resolution of 1/1000 kilograms. Weight data will not be included in the response if weight data is not available, the scale is not configured for Metric mode, or the scale is configured for a remote display but no display is detected. If the scale is not configured for Metric mode, the “Unacceptable Command” status bit (status byte 0, bit 6) shall be set.

Scale response: Send two- or three-byte status and five-digit weight data if available and allowed. See section 4.2.4 *Response Description* for details.

#### 4.2.2.3 Zero Scale

Command: **03h, 00h, 00h, 00h, 00h**

This command zeroes the scale. The scale will set it’s “no load” reading to the current weight reading. Scale response: Send two- or three-byte status . See section 4.2.4 *Response Description* for details.

#### 4.2.2.4 Enable Extended (3-Byte) Status

Command: **04h, 00h, 00h, 00h, 00h**

This command enables the scale to return a three-byte status.

Scale response: Send three-byte status . See section 4.2.4 *Response Description* for details.

Note: By default, the scale will return a two-byte status.

#### 4.2.2.5 Disable Extended (3-Byte) Status

Command: **05h, 00h, 00h, 00h, 00h**

This command enables the scale to return a two-byte status.

Scale response: Send two-byte status . See section 4.2.4 *Response Description* for details.

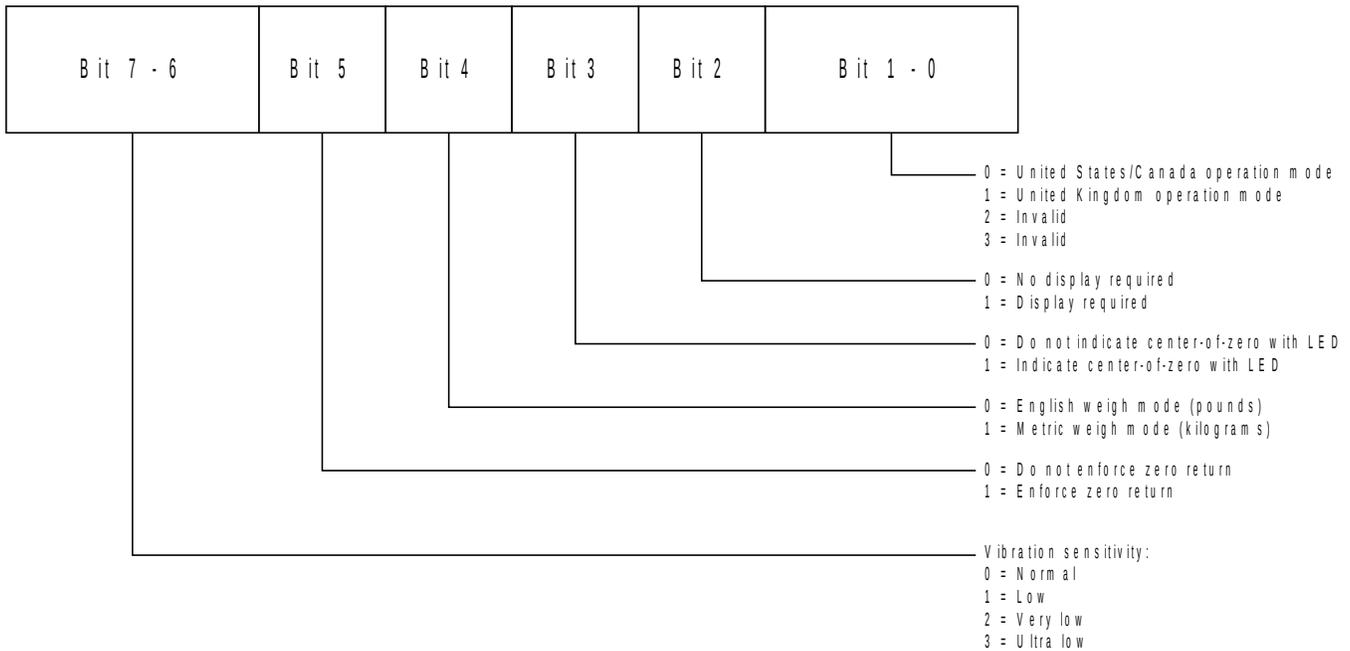
Note: By default, the scale will return a two-byte status.

#### 4.2.2.6 Configure Scale

Command: **20h, 00h, cfg0, cfg1, cfg2**

This command configures the scale Three (3) data bytes are sent in the command. These data bytes determine the scale's operating parameters. The configuration data bytes are defined as follows:

cfg0:



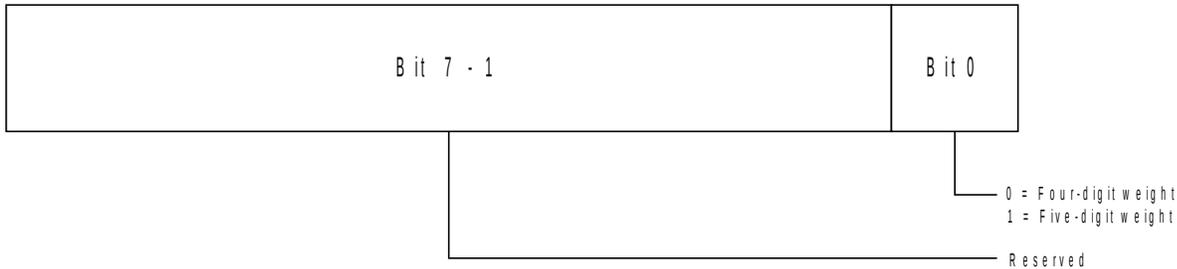
#### Operation Mode:

United States/Canada - Conforms to regulations specified by the United States NIST Handbook 44 and the Canadian Department of Consumer and Corporate Affairs, Weights and Measurements Act, Specifications SGM-1.  
 United Kingdom - Conforms to regulations specified by the Non-automatic Weighing Instruments (EEC Requirements) Regulations 1992 (based on OIML R 76-1).

**Scale Display:** If a display is required and no display is detected, the scale will respond to weight requests with a hardware error status until either a display is connected or the scale is configured for no display.

**Enforced Zero Return:** The scale shall have a configurable enforced zero return. No weight data can be transmitted to the host until a re-zeroing occurs after an enforced zero return condition.

cfg 1:



**Weight Data Length:** The weight data length (four or five digits) is only configurable when the scale is configured for English weigh mode. Five digits are always returned for Metric weigh mode.

**cfg2:** Reserved - 00h

Scale response: Send two- or three-byte status. In three-byte status indicate successful or unsuccessful configuration. See section 4.2.4 *Response Description* for details.

#### 4.2.2.7 Report Scale Configuration

Command: **21h, 00h, 00h, 00h, 00h**

This command returns the scale configuration bytes as described in section 4.2.2.6 *Configure Scale*. Three (3) data bytes are returned.

Scale response: Send two- or three-byte status indicating “Configuration data response frame” plus scale operating parameter configuration data (3 bytes). See section 4.2.4 *Response Description* for details.

#### 4.2.2.8 Clear Remote Display

Command: **06h, 00h, 00h, 00h, 00h**

This command clears the scale display if the scale is in United Kingdom operation mode. If the scale is in United States operation mode, this command has no effect.

Scale response: Send two- or three-byte status. If the scale is in United States operation mode, indicate “Unacceptable Command”. See section 4.2.4 *Response Description* for details.

#### 4.2.2.9 Test Request

Command: **00h, 10h, 00h, 00h, 00h**

This command requests that the scale perform its self-test.

Scale response: Send two- or three-byte status . See section 4.2.4 *Response Description* for details.

#### 4.2.2.10 Status Request

Command: **00h, 20h, 00h, 00h, 00h**

This command requests status from the scale.

Scale response: Send two- or three-byte status. See section 4.2.4 *Response Description* for details.

#### 4.2.2.11 System Reset

Command: **00h, 40h, 00h, 00h, 00h**

This command resets the scale.

Scale response: Refer to section 4.2.4 *Response Description* for scale response.

### 4.2.3 Response Format

1st	2nd	3rd	4th - 8th
Status Byte 0	Status Byte 1	Status Byte 2 or Data byte	Data bytes 1 - 5

where:

V The Status Byte 0 is defined as follows (bit 0 is the least significant bit):

Bit Position	Meaning
0	Flash update in progress (if flash update is implemented)
1	Configuration data response frame
2	Extended status response frame
3	Not defined - always 0
4	Not defined - always 0
5	Not defined - always 0
6	Unacceptable command
7	Device not ready to receive weigh commands

√ The Status Byte 1 is defined as follows (bit 0 is the least significant bit):

Bit Position	Meaning
0	0:English weigh mode; 1:Metric weigh mode
1	0:Four-digit weight; 1:Five-digit weight
2	Weight data not included/scale in motion
3	Data value error (weight digits not in range 0-9)
4	Read error (timeout occurred trying to obtain valid weight/status)
5	Remote display required but not detected
6	Scale hardware error
7	Undefined command received (command reject)

√ Status Byte 2 is defined as follows (bit 0 is the least significant bit):

Bit Position	Meaning
0	Configuration successful
1	Scale under zero
2	Scale over capacity
3	Scale center-of-zero
4	Scale requires zeroing
5	Scale warm-up in progress
6	Duplicate weight (United Kingdom mode only)
7	Not defined - always 0

Notes:

√ Status byte 2 is only returned when extended status has been enabled.

❖ Status byte 1, bit 2 will also be set whenever any of the following are set:

Status byte 0, bits 6 or 7

Status byte 2, bits 1 through 5

√ In status byte 2, bit 3 will **not** be set if bit 1, 2, or 4 is set.

√ In status byte 1, bit 6 will be set whenever bit 5 is set.

√ Status byte 2, bit 6 will be set when the scale is operating in United Kingdom mode and has received a weight request when the scale did not return to zero prior to the receipt of the weight request.

√ Status byte 1, bit 1 only applies to English weight requests. For all other requests it is undefined/always 0.

## 4.2.4 Response Description

The scale firmware shall pad the response to 8 bytes. Refer to section 4.3.1.1 *Report Descriptor* for details.

### 4.2.4.1 Command Reject

Response: **x0000x0xb, 1xx001xxb <, 00xxxxx0b>**

This response is returned after receiving an undefined or unsupported command.

#### 4.2.4.2 Successful Configuration

Response: **x0000x00b, 0xxxx1xxb <, 00xxxxx1b>**

This response is returned following the successful execution of a Configure Scanner command.

#### 4.2.4.3 Unsuccessful Configuration

Response: **x0000x0xb, 0xxxx1xxb <, 00xxxxx0b>**

This response is returned following the receipt of invalid parameter specifications in a Configure Scale command.

#### 4.2.4.4 Configuration Data

Response: **x0000x10b, 0xx001xxb, <00xxxxx0b,> cfg0, cfg1, cfg2**

This response is returned following the receipt of the Report Configuration command. The two- or three-byte status is returned followed by three (3) data bytes. See section 4.2.2.6 *Configure Scale* for the format of **cfg0..cfg2**.

#### 4.2.4.5 Unacceptable Command

Response: **01000x00b, 000001xxb <, 00xxxxx0b>**

This response is returned following the receipt of a command which is a valid scale command but which cannot be executed in the scale's current configuration. For example, this response would be sent if the scale receives an English Weight Request while it is configured in metric mode, the scale receives a Metric Weight Request while it is configured in English mode, or the scale receives a Clear Remote Display command while it is configured in United States/Canada operations mode.

#### 4.2.4.6 English Weight Request (Four-digit weight)

Response: **00000x00b, 0xxxxx00b, <00xxxxx0b, xxh, xxh, xxh, xxh >**

This response includes a two- or three-byte status followed by the weight data bytes, if available and allowed. Four (4) bytes of weight data are returned. The weight bytes are BCD-formatted numbers in the range of 0-9 inclusive.

Weight data will **not** be included if weight data is not available, the scale is configured for a remote display and a remote display is not detected, or some other scale hardware error is detected.

#### 4.2.4.7 English Weight Request (Five-digit weight)

Response: **0000x00b, 0xxxxx10b, <00xxxxx0b, xxh, xxh, xxh, xxh, xxh>**

This response includes a two- or three-byte status followed by the weight data bytes, if available and allowed. Five (5) bytes of data are returned. The weight bytes are BCD-formatted numbers in the range of 0-9 inclusive.

Weight data will **not** be included if weight data is not available, the scale is configured for a remote display and a remote display is not detected, or some other scale hardware error is detected.

#### 4.2.4.8 Metric Weight Request

Response: **0000x00b, 0xxxxx01b, <00xxxxx0b, xxh, xxh, xxh, xxh, xxh>**

This response includes a two- or three-byte status followed by the weight data bytes, if available and allowed. Five (5) bytes of data are returned for an English Weight Request. The weight bytes are BCD formatted numbers in the range of 0-9 inclusive.

Weight data will **not** be included if weight data is not available, the scale is configured for a remote display and a remote display is not detected, or some other scale hardware error is detected.

#### 4.2.4.9 Successful Zero Scale

Response: **0000x00b, 000001xxb, <000001000b>**

This response is returned upon successful zeroing of the scale after a Zero Scale command.

#### 4.2.4.10 Unsuccessful Zero Scale

Response: **0000x0xb, 0xx001xxb, <00xx0xx0b>**

This response is returned if the scale could not be zeroed for a Zero Scale command.

#### 4.2.4.11 Enable Extended Status

Response: **0000010xb, 0xx001xxb, 00xxxxx0b**

This response is returned for an Enable Extended Status command.

#### 4.2.4.12 Disable Extended Status

Response: **000000xb, 0xx001xxb**

This response is returned for a Disable Extended Status command.

#### 4.2.4.13 Successful Clear Remote Display

Response: **0000x00b, 0x0001xxb<, 00xxxxx0b>**

This response is returned following the successful execution of a Clear Remote Display command.

#### 4.2.4.14 Unsuccessful Clear Remote Display

Response: **0000x0xb, 0xx001xxb<, 00xxxxx0b>**

This response is returned following the receipt of a Clear Remote Display command when the scale is not in United Kingdom operation mode.

#### 4.2.4.15 Status Request

Response: **x0000x0xb, 0xx001xxb <, 00xxxxx0b>**

This response is returned after receiving a Status Request command.

#### 4.2.4.16 System Reset

There is no response for the Reset command. However, the scale shall initialize, perform basic assurance test (BAT), and re-enumerate.

#### 4.2.4.17 Test Request

Response: **x0000x0xb, 0xx001xxb <, 00xxxxx0b>**

This response is returned after receipt of a Test Request command.

## 4.3 Firmware Consideration

The scale shall use a Vendor Defined Page 1 (FF45h) usage page for all of its usages. There are 3 usages defined for the scale in this page:

V Scale: 6E00h  
V Scale Command: 6E01h  
V Scale Status: 6E02h

### 4.3.1 Descriptors

---

Device, configuration, interface, and endpoint descriptors are device specific.

#### 4.3.1.1 Interface String Descriptor

The flash interface string descriptor shall be as follows:

**Scale (Usage = 6E00h, Usage Page = FF45h)**

#### 4.3.1.2 Report Descriptor

The scale shall have one output and one input reports to send commands and receive status/data respectively. The output report will have a fixed length of 5 bytes. The input report will have a fixed length of 8 bytes. Under the HID device model, output reports will be sent out using a control write on end point 0. Input reports will be sent in on an endpoint other than zero (device specific).

The report descriptor for the scale shall be as follows:

Usage Page (Vendor Defined Page 1)  
Usage (Scale)  
Collection (Application)  
  Usage (Scale Command)  
  Report Size (8)  
  Report Count (5)  
  Logical Minimum (0)  
  Logical Maximum (255)  
  Output (Data, Var, Abs)  
  Usage (Scale Status)  
  Report Count (8)  
  Input (Data, Var, Abs)  
End Collection

## 5 Pseudo-RS232 Interface

### 5.1 Introduction

The information provided in this section is intended to facilitate the development of USB devices for which today there exists an "RS232" device which attaches to the RS485 device channel on IBM Point-of-Sale (POS) machines. **The interface described here is not intended to be a general purpose RS232-like interface for a USB device.** The goal of providing this interface is to protect our customers' application investment by providing USB hardware which will work without modifications to existing applications.

### 5.2 Command/Response Set

#### 5.2.1 Command Format

The Pseudo-RS232 device command is fixed length (251 bytes) and is laid out as follows:

1st-2nd	3rd	4th	5th-251st
Length	Command byte 1	Command byte 2	Up to 247 bytes of data

where:

Command byte 1	Command byte 2	Pseudo-RS232 command
00h	10h	Test
00h	20h	Status request
00h	40h	Reset
01h	00h	Setup
02h	00h	Channel Control
04h	00h	Read Status
08h	00h	Transmit
10h	00h	Receive Buffer

Data bytes are used for the Setup, Channel Control, and Transmit commands. Length is a two-byte value (little endian, Intel format) and is the total length of the command including the command bytes and the length itself.

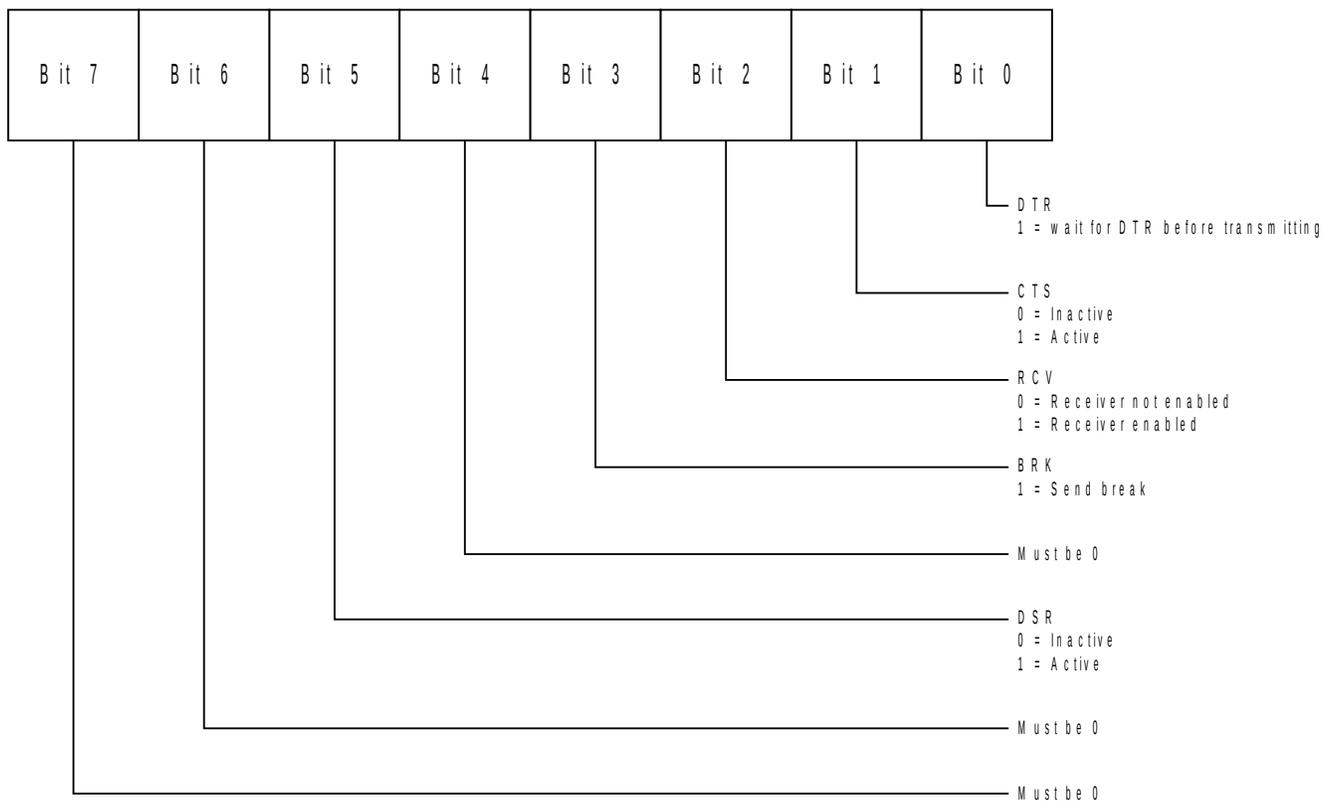
## 5.2.2 Command Description

### 5.2.2.1 Setup

Command: **0Ah, 00h, 01h, 00h, 00h, 00h, ctrl, 00h, 00h, 00h**

This command sets the initial (RS232-like) state of the device. The RS232 state is described by the **ctrl** byte which is formatted as shown below.

ctrl:



Device response: Send two-byte status. See section 6.2.4 *Response Description* for details.

### 5.2.2.2 Channel Control

Command: **05h, 00h, 02h, 00h, ctrl**

This command establishes the (RS232-like) state of the device. The format of **ctrl** is as shown in the Setup command. This command has the same effect as the Setup command.

Device response: Send two-byte status indicating new device state. See section 5.2.4 *Response Description* for details.

### 5.2.2.3 Read Status

Command: **04h, 00h, 04h, 00h**

This command requests status from the device. This command has the same effect as the Status Request command.  
Device response: Send two-byte status . See section 6.2.4 *Response Description* for details.

### 5.2.2.4 Transmit

Command: **length, 08h, 00h, data1, ... , dataN**

This command is used to send data to the device.

**length** (2 bytes): Maximum 251

**data1, ... , dataN**: A maximum of 247 bytes of data can be sent in a single transmit command. This data is passed “as is” from the application to the device; the driver does no modification/interpretation of the data.

Device response: Send two-byte status . See section 6.2.4 *Response Description* for details.

### 5.2.2.5 Receive Buffer

Command: **04h, 00h, 10h, 00h**

This command forces the device to send data.

In normal operation, when the device has data to send, it will send status to the driver indicating that it has data; the driver will then request that data from the device (via feature report which is accomplished via control read) and will then notify the application that data is available. In this case, the driver asks for data rather than being told that the device has data.

Device response: Send two-byte status and any available data. See section 6.2.4 *Response Description* for details.

### 5.2.2.6 Test Request

Command: **04h, 00h, 00h, 10h**

This command requests that the device perform its self-test.

Device response: Send two-byte status . See section 6.2.4 *Response Description* for details.

### 5.2.2.7 Status Request

Command: **04h, 00h, 00h, 20h**

This command requests status from the device.

Device response: Send two-byte status. See section 6.2.4 *Response Description* for details.

### 5.2.2.8 System Reset

Command: **04h, 00h, 00h, 40h**

This command resets the device.

Device response: Refer to section 6.2.4 *Response Description* for scale response.

## 5.2.3 Response Format

1st	2nd
Status Byte 0	Status Byte 1

where:

∨ The Status Byte 0 is defined as follows (bit 0 is the least significant bit):

Bit Position	Meaning
7	Break in progress
6	Timer overflow
5	0 = Receive buffer 1 empty
4	0 = Receive buffer 2 empty
3	Flash update in progress (if flash update is implemented)
2	Overrun occurred at the end of this buffer
1	Receive buffer 1 full
0	Receive buffer 2 full

∅ The Status Byte 1 is defined as follows (bit 0 is the least significant bit):

Bit Position	Meaning
7	0=Transmit buffer empty; 1=Transmit buffer not empty
6	Initialized (Setup command has been processed)
5	Command error
4	Reserved
3	Soft error on bring-up
2	RTS (Ready To Send)
1	DTR (Data Terminal Ready)
0	Receive buffer data error

All bits are active high unless specified otherwise.

## 5.2.4 Response Description

---

The device firmware always returns two bytes of status. Refer to section 6.3.1.1 *Report Descriptor* for details.

### 5.2.4.1 Command Reject

Response: **00xx0xxb, 00100xx0b**

This response is returned following the receipt of an invalid command.

### 5.2.4.2 Setup

Response: **x0xx00xxb, 01000xx0b**

This response is returned following the receipt of a Setup command.

### 5.2.4.3 Channel Control

Response: **x0xx00xxb, 00000xxb**

This response is returned following the receipt of the Channel Control command. The two-byte status indicates the new (RS232-like) state of the interface.

### 5.2.4.4 Read Status/Status Request

Response: **x0xx00xxb, x0000xxb**

This response is returned following the receipt of a Read Status (device) command or Status Request (system) command.

### 5.2.4.5 Transmit

Response: **00xx00xxb, 10000xx0b**

This response is returned following the receipt of a Transmit command. For the “old” RS485-attached devices, this status (with the “transmit buffer not empty”) bit set was sent when the Transmit began and another status (with the “transmit buffer not empty”) bit cleared) was sent after the data was transmitted. See 6.2.4.6 *Transmit Complete Response* below.

### 5.2.4.6 Transmit Complete

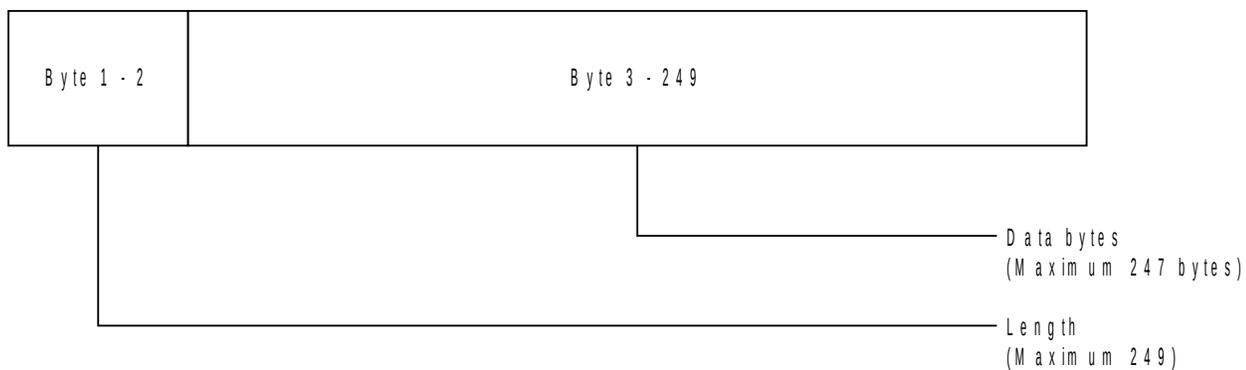
Response: **00x00xxb, 00000xx0b**

This response is returned following the completion of a Transmit command. See 6.2.4.5 *Transmit Command Response* above for details.

### 5.2.4.7 Device Data

Response: **001x00xxb, 00000xx0b**

This response is sent by the device when it has data to send. The data itself will be requested by the driver. The format of the data is as shown below.



Length is a two-byte field and shall be in the little endian format.  
See 6.3.1.1 *Report Descriptors* for details about input and output reports.

### 5.2.4.8 Receive Buffer

Response: **001x00xxb, 00000xx0b**

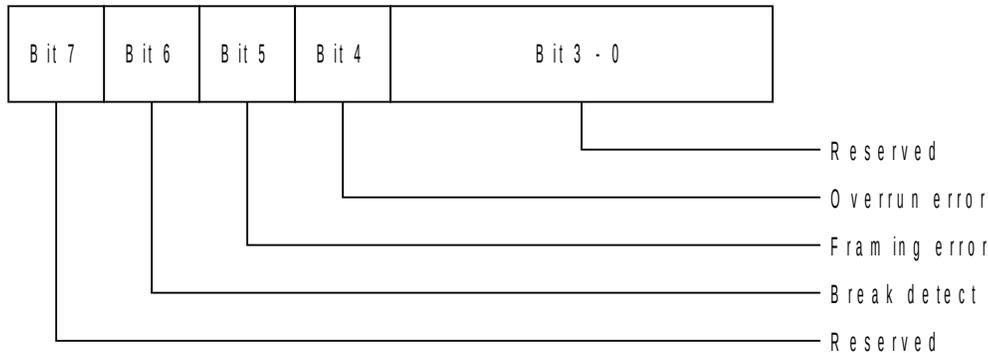
This response is returned following the receipt of a Receive Buffer command. This forces the device to return data for a partially complete command. The data format is as shown in 6.2.4.7 *Device Data Response*.

See 6.3.1.1 *Report Descriptors* for details about input and output reports.

### 5.2.4.9 Receive Buffer (Data Buffer Error)

Response: **001x00xxb, 00000xx1b**

This response is returned following the receipt of a Receive Buffer command in which there is a data error. The actual data will be requested by the driver. The data format is as shown in 6.2.4.7 *Device Data Response*. The last byte of the data shall have the following format when the “Data buffer error” status bit is set:



See 6.3.1.1 *Report Descriptors* for details about input and output reports.

#### 5.2.4.10 Reset

There is no response for the Reset command. The device performs a “soft” reset and re-enumerates.

#### 5.2.4.11 Test Request

Response: **0000000b, 00x00xx0b**

This response is returned after receipt of a Test Request command. The “command error” bit is set if the test was unsuccessful.

## 5.3 Firmware Consideration

The Pseudo-RS232 devices shall use a Vendor Defined Page 1 (FF45h) usage page for all of its usages. Followings are the usages defined for these devices in this page:

- ✓ Pseudo-RS232 device compatible with SIO address 64h: 6400h
- ✓ Pseudo-RS232 device compatible with SIO address 65h: 6500h
- ✓ Pseudo-RS232 device compatible with SIO address 68h: 6800h
- ✓ Pseudo-RS232 device compatible with SIO address 69h: 6900h
- ✓ Pseudo-RS232 device command: 6401h
- ✓ Pseudo-RS232 device status: 6402h
- ✓ Pseudo-RS232 device read data: 6403h

The command, status, read data usages are shared to all devices of this type. It is the Vendor's responsibility to provide a mechanism to switch to different device usage if the SIO address collides at the application layer. The device shall re-enumerate after the usage switch to alert the system software of the change. See section 2.2.1 *Device Enumeration Without Hot Plugging*.

## 5.3.1 Descriptors

---

Device, configuration, interface, and endpoint descriptors are device specific.

### 5.3.1.1 Interface String Descriptor

The flash interface string descriptor shall be as follows:

**Pseudo-RS232 device (Usage = 6400h, Usage Page = FF45h)**

If the device is compatible with SIO address 64h.

### 5.3.1.2 Report Descriptor

The pseudo-RS232 device shall have one output, one input, and one feature report to send commands, receive status, and receive data respectively. The output report will have a fixed length of 251 bytes. The input report will have a fixed length of 2 bytes. The feature report will have fixed length of 249 bytes. Under the HID device model, output reports will be sent using a control write on endpoint 0. Feature reports will be sent and receive using control write and control read on endpoint 0. Input reports will be sent in an endpoint other than zero (device specific).

The report descriptor for the pseudo-RS232 device shall be as follows:

Usage Page (Vendor Defined Page 1)  
Usage (Pseudo-RS232 device compatible with SIO address 6Xh)  
Collection (Application)  
  Usage (Pseudo-RS232 device command)  
  Report Size (8)  
  Report Count (251)  
  Logical Minimum (0)  
  Logical Maximum (255)  
  Output (Data, Var, Abs)  
  Usage (Pseudo-RS232 device status)  
  Report Count (2)  
  Input (Data, Var, Abs)  
  Usage (Pseudo-RS232 device read data)  
  Report Count (249)  
  Feature(Data,Var,Abs)  
End Collection

