C005066

Interface Configuration Document For:

CR900FD
CR1000
CR1000XHD
CR1400
CR1400XHD
CR1428
CR2300
CR2600
CR2600XHD

CR3600
CR3600DPM
CR4405
CR5000
CR5000RTC
CR6000
CR8000
T500

Firmware build 1206 (for CR5000 & CR5000RTC)
Firmware build 5040 (for the T500)
Firmware build 1174 (for all others)

Table of Contents

1 Keyword Table

These keywords are used throughout the document to show relationships between settings.

Keyword	Description
#2Of5	All 2 of 5 symbologies
#AES	Settings or commands related to AES encryption
#AgeVerification	Settings related to the Age Verifications functions of some Code readers
#AIMId	Settings controlling the output of the AIM Identifier
#Aztec	Aztec symbology
#Battery	Settings for battery usage
#Beep	Settings that affect beep duration and intensity
#ButtonStayDownTime	Settings that affect how long the buttons stay engaged after being pressed
#Cellphone	Settings related to cellphone reading
#Charger	Settings related to a charger base
#Charging	Settings related to charging a reader and/or phone
#Codabar	Codabar symbology
#Code39	Code 39 symbology
#Communications	Used in changing the communication mode of the reader
#CompositeBarcodes	Settings that affect reading of barcodes with more than one part
#DataMatrix	Data Matrix symbology
#DataRetention	Settings that determine the behavior of data saved on the reader
#DataEncoding	Settings that affect incoming/outgoing data
#DefaultEvent	Any settings that may affect default event behavior
#DPM	Settings related to Direct Part Mark
#DuplicateBlock	Settings related to blocking duplicate barcodes
#EAN/JAN	EAN/JAN symbology
#Encryption	Settings related to Bluetooth® encryption
#Foi0	Field of Interest 0
#Foi1	Field of Interest 1
#GoodReadRTS	Settings that affect the good read output on RTS
#GS1Databar	The GS1 DataBar family of symbologies
#HanXin	Han Xin symbology
#InterCharacterDelay	Settings controlling the USB keyboard inter-character delay
#Interleaved2Of5	Interleaved 2 of 5 symbology
#KeyboardMap	Settings related to changing the output keyboard map
#MotionDetection	Settings that affect the behavior of the motion detection feature
#PDF417	PDF417 symbology
#PharmaCode	PharmaCode symbology
#PictureSettings	Settings that affect pictures (JPG or PGM) captured by the reader
#Postal	Postal symbologies
#QR	QR Code symbology
#ReaderState	Settings that affect the transition from one state to another (i.e. Active to Idle)

#SerialComm	Settings related to serial communications mode
#SettingsLock	Settings that affect the enhanced Settings Lock feature
#SimpleAV	Settings that affect the Simple Age Verification feature on the CR5000
#Stand	Settings that affect reader behavior in a Stand
#StandDetect	Settings related to detecting the Stand, or table, or the charging base as a Stand
#StandModes	Settings that change based on entering or leaving the Stand/table
#T500	Information and settings that relate to the T500 Reader Accessory
#Telepen	Telepen symbology
#TextCommands	Settings relating to the reader accepting text commands
#UPC	UPC symbology
#Vibration	Settings that affect vibration in readers with a vibration motor

2 Scope

This Interface Configuration Document (ICD) specifies the communication protocol between the Code Reader™ 900FD (CR900FD), Code Reader™ 1000 (CR1000), Code Reader™ 1000 XHD (CR1000XHD), Code Reader™ 1400 (CR1400), Code Reader™ 1400 XHD (CR1400XHD), Code Reader™ 1428 (CR1428), Code Reader™ 2300 (CR2300), Code Reader™ 2600 (CR2600), Code Reader™ 2600 XHD (CR2600XHD), Code Reader™ 3600 (CR3600), Code Reader™ 3600 DPM (CR3600DPM), Code Reader™ 4405 (CR4405), Code Reader™ 5000 (CR5000) , Code Reader™ 5000 RTC (CR5000RTC), Code Reader™ 6000 (CR6000), Code Reader™ 8000 (CR8000), or Code T500 Reader Accessory (T500) hardware and application software that runs on the Host computer, specific Reader commands, examples of a variety of ways to communicate and send data to the Reader (i.e. RS232, USB) and command/communication types.

Make sure to use the latest released firmware for default values listed in Section 9.

3 Notations

The interface protocol is described as a set of grammars, indicated by different type styles and symbols. These indications are listed in the table below.

Example	Indication	Grammar
Text-Command	Italic type	Syntactic categories (non-terminals)
space	Bold type	Terminal symbols
0xFF	0x prefix indicating hexadecimal	Literal byte values
'X'	Single quotes	Literal ASCII characters
SOH	All caps	Non-printable ASCII characters
PageUp	Key name	key press-release sequence
shift↓	Key plus down arrow	Key-down only
shift↑	Key plus up arrow	Key-up only
esc | tab	Vertical bar	Alternatives (this or that)
data$_{opt}$	$_{opt.}$ (opt subscript)	Optional terminals and non-terminals
packet-type$_{nz}$	$_{nz}$ (nz subscript)	Applies to all packets except *z* type packets
crc16$_{nr}$	$_{nr}$ (nr subscript)	Applies to packets sent in "non-raw" mode, i.e. in "packet" mode

4 Reader Command Overview

This section is intended to introduce developers/users to the methods used to send commands to the Reader. There are two ways to send a command to the Reader: from a Host computer, or by scanning a barcode containing a command sequence. When sending commands from the host, there are two formats that can be used: packetized commands and text commands.

4.1 Packetized Commands

Packetized commands are the most reliable way to communicate to the Reader. The packet consists of a prefix and a suffix. The prefix contains the amount of data to be transmitted and the suffix contains error detection. Unlike text commands, packetized commands are always enabled. (See Section 7.2)

4.2 Text Commands

Text commands are provided as an easy way to send a command to a Reader but they lack the reliability of packetized commands. In addition, text commands must be enabled. Text commands can easily be sent from a terminal program and use a %xx (similar to URL encoding) to translate an escape sequence containing a 2-digit hex value corresponding to the single 8-bit ASCII character. This allows non-printable ASCII characters to be entered via the terminal program. Text commands can be sent via the RS232, USB Virtual COM (VCOM) or USB HID mode by using appropriate communication software.

4.3 Code Reader Batch (CRB) System

The Code Reader Batch (CRB) system is a convenient method for creating and maintaining a set of commands that can be sent to the Reader. The commands in the CRB file are immediately processed by the reader.

These CRB files can be created in any text editor with the file extension of .crb. The CRB system accepts all of the valid *text commands*. There should be one command per line. The CRB file may contain empty lines and comments (prefixed with the semicolon (;) character) as well.

The CRB files can be sent directly to the Reader using the normal file transfer. As CRB files are just a list of *text commands*, they can also be sent by a serial terminal program. **Note: if using a serial terminal program the Reader will first need to be commanded in to "text command mode"; see Section 7.1.**

CRB files can also be encoded in barcode form using Code's CortexTools® utility to be processed by scanning the barcode with a Code reader.

Pre-made Configuration barcodes can be found on Code's web site (http://codecorp.com), and configuration CRB files can be requested via support@codecorp.com.

4.4 Barcode Commands

The Reader will recognize the following sequence within a barcode as a command to the Reader:

SOH 'X' GS STX *Text-Command* **EOT** (Packet does not contain spaces)

The *Text-Command* portion contains a text command as described above.

Because the Barcode Command is terminated with ASCII EOT, the *Text-Command* may not contain EOT. If the *Text-Command* needs to contain EOT, encode it as %04. Multiple commands can be contained in a single barcode.

5 Communication Medium

The Reader communicates with the Host via USB (keyboard/HID/VCOM), RS232, or Bluetooth®. The Host includes appropriate hooks and/or drivers to enable two-way communication with the Reader.

Note: USB keyboard communications are one-way, from the Reader to Host only. A special sequence is available to switch the Reader from keyboard to HID communication mode. See register 1B.

5.1 USB Enumeration PID

The following Product ID (PID) are used by Code to identify their products when connected to a PC. All Code products use the same Vendor ID (VID) of 11FA

PID	Product family	Description
0x0200	Reader	Keyboard
0x0201	Reader	IBM POS (also known as IBM Handheld)
0x0202	Reader	HID Native (also known as Downloader)
0x0204	Reader	Code VCOM
0x0205	M3 Modem	Keyboard
0x0206	Reader	HID POS
0x0207	M3 Modem	DFU
0x0208	M3 Modem	HID POS
0x0209	M3 Modem	Keyboard + bulk interface
0x020A	T500 Cable	Keyboard (Used to communicate with CortexTools®)
0x0214	M3 Modem	Code VCOM + HID
0x0215	Reader	CDC
0xFFFF	M3 Modem	Modem bootloader

6 Reader to Host Communication

The Reader may be configured in raw mode, where no packet framing or check characters are sent, and packet mode (See sections 4.1 and 4.2) The Reader may also be configured to expect an acknowledgement from the Host after each packet and automatic retry when no acknowledgement is received. Standard “one-way” mode of operation uses raw packets, no expected response from Host, and no automatic retry. Standard “two-way” mode of operation uses packets with framing and checks characters, expects a response from the Host, and automatically resends. If no Acknowledgement is received (ACK), three (3) attempts to resend are made.

6.1 Raw Data

Reader to Host communication consists of decoded raw data having no framing or check characters. Raw data is sent with no “end of packet” data (crc16), expects no response from Host and no data is resent.

6.2 Packet Data

Data from the Reader to the Host consist of *packets* as specified below. Packetized data is sent using ACK/NAK protocols with framing and check characters. Packets are delivered asynchronously as barcodes are read and in response to Host to Reader commands. For keyboard communication (USB keyboard), all *ASCII-character*s are transmitted as *keyboard-sequence*s. For all other communication ports, all *ASCII-character*s are transmitted as *ASCII-byte*s.

Note: Even though the data size field allows up to 65535 bytes of data in a packet, the actual size of a packet either in raw or in packet mode including data and packet overhead is a maximum of 16384 bytes.

Note: The following acronyms from the ASCII character definitions are used below:

- SOH – Start of Heading (0x01)
- STX – Start of Text (0x02)
- EOT – End of Transmission (0x04)
- RS – Record Separator (0x1E)

Language Part	Definition
packet	*start packet-type*$_{nz}$ *data*$_{opt}$ *end*
start	*packet-start*$_{nr}$ **|** *codeXML-start*$_{nz}$
packet-start	**SOH ‘X’ ‘R’** *protocol-version Reader-id packet-number timestamp data-size*
protocol-version	**‘1’**
Reader-id	big-endian 32-bit number
packet-number	*data-packet-number* | *cmd-packet-number*
data-packet-number	any byte value in the range [0,7f]; increments with each packet; does not increment with resends; used with *z* and *a* packets only
cmd-packet-number	any byte value in the range [80-ff]; increments with each packet; does not increment with resends; used with all packets other than *z* and *a*
timestamp	big-endian 32-bit number, indicates timestamp in seconds (relative to Reader power-up) (For all but *z* packets, the timestamp represents the time the packet was sent to the Host; for *z* packets, the time the barcode was read.)
data-size	big-endian 16-bit number indicating size of the *data* field (in bytes)
codeXML-start	**SOH ‘X’ RS** *tag_response* **‘/’**
tag_response	**‘ap’**
packet-type	Single ASCII-character in table below

<table>
<tr><td><i>data</i></td><td><i>character</i> | <i>data character</i></td></tr>
<tr><td><i>character</i></td><td><i>byte</i>| <i>keyboard-sequence</i></td></tr>
<tr><td><i>byte</i></td><td>any byte value in range [<b>0x00</b>,<b>0xFF</b>]</td></tr>
<tr><td><i>keyboard-sequence</i></td><td><i>key</i>| <b>shift↓</b> <i>key</i> <b>shift↑</b>| <b>alt↓</b> <i>decimal-code</i> <b>alt↑</b></td></tr>
<tr><td><i>key</i></td><td><b>~ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | - | = | q | w | e | r | t | y | u | i | o | p | [|] | \| a | s | d | f | g | h | j | k | l | ; | ’ | z | x | c | v | b | n | m | , | . | /| space| esc | tab | shift | alt | ctrl | enter | backspace| f1 | f2 | f3 | f4 | f5 | f6 | f7 | f8 | f9 | f10 | f11 | f12| insert | delete | home | end | pageup | pagedown| left | right | up | down | keypadenter|</b> <i>digit</i></td></tr>
<tr><td><i>decimal-code</i></td><td><i>digit</i> | <i>digit digit</i> | <i>digit digit digit</i> (range [0,255])</td></tr>
<tr><td><i>digit</i></td><td><b>keypad0</b> | <b>keypad1</b> | <b>keypad2</b> | <b>keypad3</b> | <b>keypad4</b>| <b>keypad5</b> | <b>keypad6</b> | <b>keypad7</b> | <b>keypad8</b> | <b>keypad9</b></td></tr>
<tr><td><i>end</i></td><td><i>codeXML-end</i>$_{nz}$ <i>crc16</i>$_{nr}$</td></tr>
<tr><td><i>codeXML-end</i></td><td><b>EOT</b></td></tr>
<tr><td><i>crc16</i></td><td><i>big-endian 16-bit number</i> representing crc16 of the packet, calculated over the entire packet, excluding the crc16 itself. See source files <i>crc16.[hc]</i> in
Appendix: Example CRC16 C Code for details on the crc16 algorithm and polynomials to be used.</td></tr>
</table>

The following *packet-types* are defined:

a	Append decode data; indicates that *data* contains the first part of the decode data. A sequence of *‘a’* packets always ends with a *‘z’* packet. The data of all *‘a’* packets in a group and the final *‘z’* packet should be concatenated by the Host.
d	Done response; command and its associated data were successfully received; *data* optionally contains a null-terminated text message.
e	Error response; command was not successfully received; *data* optionally contains a null-terminated text message.
g	Start of a group of *‘z’* packets to follow, terminated by a *‘d’* or *‘e’* packet (*‘d’* for complete group, *‘e’* for incomplete group)

i	indicates that data contains the zero-terminated Reader information string (of printable ASCII characters and TAB) in the following format: iVVVVWWWWXXXXSSSSSSSSSSAOODYYYYHHIIIIJJJJKKKKLLLL<TAB>Z…Z where: i indicates 'I' string output VVVV is the application firmware version number; WWWW is the core application firmware version number; XXXX is reserved; SSSSSSSSSS is the Reader's serial number (ten digits); A is the current execution state: "A" means core is running OO is the OEM identifier; D is the display type: "0" is no display device. YYYY is reserved; HH is the hardware revision; IIII is the hardware type identifier (value in register 21B); JJJJ is the boot application version; KKKK is the operating system kernel version; LLLL is the root file-system version; <TAB> is the ASCII TAB character; Z…Z is the OEM decoder version: a null terminated string of printable ASCII characters.
m	Message response; *data* contains a message (comment). *'m'* packets are not sent when the Reader is in "raw" mode.
r	Read barcode failure; decoder attempted but failed to read a barcode.
z	Decoded data from a barcode; *data* contains the data decoded from the barcode.

In "raw" mode (as opposed to "packet" mode), type **m** packets are not sent, only the decoded data is sent for type **z** packets, and all other packets are sent without the *packet-start* and *crc16* fields. In "packet-mode," the *packet-start* and *crc16* fields are always sent. (See Figure 1)

Raw Mode						
‘z’ (data) packet:						
Data						
*(non-z) packet:example ‘**i**'*						
CodeXML® ‘**i**’ response example						
CodeXML-start				packet-type	data	CodeXML-end
SOH	**‘X’**	**RS**	**‘ap/’**	**‘i’**	VVV…	**EOT**

Packet Mode															
‘z’ (data) packet:															
packet-start								Data							packet-end
SOH	**‘X’**	**‘R’**	**‘1’**	Reader ID (4 bytes)	packet number (1 byte)	time stamp (4 bytes)	data size (2 bytes)	data							crc16 (2 bytes)
*(non-z) packet:example ‘**i**’*															
packet-start								CodeXML® ‘**i**’ response							packet-end
SOH	**‘X’**	**‘R’**	**‘1’**	Reader ID (4 bytes)	packet number (1 byte)	time stamp (4 bytes)	data size (2 bytes)	**SOH**	**‘X’**	**RS**	**‘ap/’**	**‘i’**	VVV…	**EOT**	crc16 (2 bytes)

Figure 1: Example 'z' and 'i' Packets in Raw and Packet Modes

Optionally, whenever the Host receives a packet, the Host will respond by sending a *Y* or *R* packet *(defined in the Host to Reader Communication section)* to the Reader. If the ‘expect response’ option is enabled in the Reader configuration, the Reader will repeatedly retransmit the packet (a configurable number of times) until it receives a *Y* packet.

If a packet received by the Host has a *packet-type* that is not any of the valid types listed above or has the same *packet-number* as the last processed packet of the corresponding type (command or data), the entire packet – up to and including *end* or until timeout – should be discarded by the Host. If the Host had requested a response, it should reissue the request.

If a packet received by the Host from the Reader fails its CRC, the Host should send an *R* packet to the Reader to request that the packet be resent.

7 Host to Reader Communication

Commands and data from the Host to the Reader are sent in the form of *commands* as specified in this section.

Commands are normally sent in USB Native, VCOM, or RS232 modes. Commands may not be sent via keyboard modes.

Two command formats are supported: **text-command** and **packetized-command**. Text-command format is applicable to RS232 and VCOM modes but only if the Reader is configured to accept this format. Packetized-command format is applicable to all interfaces.

text-command: See Section 7.1.

packetized-command: See Section 7.2.

The command types are explained in Section 7.3.

After the Host sends each complete command, it should wait for a response packet from the Reader. Expected responses are specified along with the command types in section 7.3. If the Reader responds with an *'e'* packet or doesn't respond within a reasonable timeout period, the Host should resend the command a reasonable number of times.

7.1 Text Commands

Keywords: #TextCommands

Text commands may be sent to the Reader in RS232 or VCOM mode using any serial communications software (e.g., HyperTerminal).

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

Language Part	Definition
text-command	*command-type encoded-data*$_{opt}$ *carriage-return*
command-type	Single ASCII character in the set defined in Section 7.3
encoded-data	*encoded-datum | encoded-data encoded-datum*
encoded-datum	*printable-character |* **%** *hex-digit hex-digit*
printable-character	any byte value in the range [0x20,0x7E]
hex-digit	**'0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'| 'A' | 'B' | 'C' | 'D' | 'E' | 'F'| 'a' | 'b' | 'c' | 'd' | 'e' | 'f'**
carriage-return	**0x0D**

In order to eliminate inadvertent commanding of the Reader, Text Commands are disabled by default.

To enable Text Commands requires an initial sequence: *;>PAx* where *x* is as defined in section 9, register 41. (**Note:** 'A' is the ASCII character that corresponds to 41 HEX.)

For example, to send the Reader commands by typing commands in HyperTerminal:

```
;>PA1
P(xx)yy
P(xx)yy
W
PA8
```

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

Note: ;>PA1 is used for interactive text commands. If the commands are to be saved in a file and sent non-interactively, use **;>PA7** instead; this enables text commands but disables echo and command responses. (See Section 7.3, Section 9, and Section 10 for additional information.)

With text commands enabled, the following two examples can be sent to a Reader in RS232 mode from HyperTerminal by just typing the example text.

Example 1 - Make the Reader beep/vibrate 3 times (**Note:** Readers with a vibration motor are the CR1400, CR1400XHD, CR1428, CR2600, CR2600XHD, CR3600, CR3600DPM and CR6000.):

#%03 *Expected output: should make Reader beep/vibrate 3 times*

Example 2 - Set Reader to continuous-read, High Density field (FOI 0) only:

P(C4)5 *Expected output: should set Reader to continuous-read, High Density field (FOI 0) only*

7.2 Packetized Commands

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

Language Part	Definition
normal-command	*prefix command-type data-size data$_{opt}$ reserved crc14*
prefix	**0xEE 0xEE 0xEE 0xEE**
command-type	Single ASCII character in the set defined in section 7.3
data-size	byte value in range [0,240], which indicates size of *data* (in bytes)

data	*datum | data datum*
datum	any byte value in the range [0,255]
reserved	**0x00**
crc14	Two consecutive bytes, each in range [0,127], representing the crc16 value & with the value 0x7F7F, most significant byte first. The packet crc16 is calculated over the entire packet, excluding the *prefix* and the crc14 itself. (See source files in Appendix: Example CRC16 C Code and Appendix: Example CRC14 C Code for details on the crc16 algorithm and polynomials to be used, as well as how to implement appropriately for crc14 transmission.)

7.3 Command Types

#	Causes the Reader to beep and vibrate the specified number of times; *data* contains the number as a single character in the range [0,127]. (The Reader will respond with *d* or *e*.) Example – beep three times: #%03 **Note:** Behavior is defined per platform. CR900FD/CR1000/CR1000XHD/CR2300/CR5000/CR5000RTC/CR8000 – Beep three times. CR1400/CR1400XHD/ CR1428/CR2600/CR2600XHD/CR3600/CR3600DPM/ CR6000 – Beep and vibrate three times. **Note:** Readers with a vibration motor are the CR1400, CR1400XHD, CR1428, CR2600, CR2600XHD, CR3600, CR3600DPM and CR6000.
$	Posts an event to the Reader; *data* contains the event number as a single character. See register C4 in Section 9 for a list of the event numbers. (The Reader will respond with *d* or *e*.) **Keywords:** #DefaultEvent
(	Causes the Reader to upload any logged error messages (no *data*) (The Reader will respond with a *g* packet, zero or more *z* packets, and a final *d* or *e*. Each *z* packet contains a portion of the requested data in its *data* field. **Note:** This is very similar to the response to the X command; however, *p* packets are not applicable and the *g* and *d*/*e* packets are not suppressed even in raw mode.)
)	Causes the Reader to erase its log of error messages (no *data*) (The Reader will respond with *d* or *e*.)
,	Causes the Reader to send a list of current Reader settings (no *data*) (The Reader will respond with *d* containing a space-separated list of all setting values (in order, expressed as hexadecimal ASCII characters) or with *e*.)

	(see ‘<’ command for saved Reader settings)
/	Toggle a bit (or bits) in a Reader setting; *data* contains a printable ASCII string in the following format: hexadecimal register number in parentheses followed by a 32-bit signed integer value, expressed in ASCII hexadecimal characters (with optional minus sign) or ASCII decimal characters preceded by the ‘#’ character, e.g., /(2e)1000 or /(2e)#4096; the specified integer is XOR’ed with the existing setting value. The way a / command is handled is equivalent to a P command – the effects are immediate but won’t survive a reboot. If you want the setting to be set after a reboot, issue a / then W or use the newer C/ combination. (The Reader will respond with *d* or *e*.) **Note:** see Section 9 for possible Reader settings.
1	Indicates the start of a file download; *data* is empty. This command is followed by a sequence of *2* commands containing the file data and a download-end command (e.g., *5*). (The Reader will respond with *d* or *e*.)
2	Indicates a continuation of a file download; *data* contains the next portion of the file data. (The Reader will not send any response.)
5	Indicates the end of a regular file download; *data* contains the name of the file, which is from 1 to 200 letters, digits, periods, hyphens, and underscores, terminated with ASCII NUL. (The Reader will respond with *d, e,* or *f*.)
9	Requests the Reader to delete a file from its storage; *data* contains the file name, terminated with ASCII NUL.
;	Reserved (no operation – treated as a comment)
<	Causes the Reader to send a list of saved Reader settings (no *data*) (The Reader will respond with *d* containing a space-separated list of all setting values (in order, expressed as hexadecimal ASCII characters) or with *e*.)(see ‘,’ command for current Reader settings)
=	Puts setting directly to Reader’s non-volatile memory so that it will take effect upon next reboot; *data* is as defined in the */* command; the specified integer replaces the existing setting value. **Note:** this command can be used to set communication modes without losing communication during the process. (The Reader will respond with *d* or *e*.) **Note:** The = command does not save changes immediately but those settings will be applied *after* a reboot. If you issue an =(2B)0 command to change the value of register 2B from 1, then until you reboot the value of 2B will still be 1. After you reboot the reader, the value of the 2B register will be 0 when the reader is ready again. In order to have that setting be immediate and survive a reboot you must either issue a P then a W command or use the newer C command. C(2B)0 is the same as these two commands combined: P(2B)0, W

	The = command is most useful to 'pre-set' a new communication mode that will become active *after* a reboot. (The Reader will respond with d or e.) **Note**: Also see commands O, P, Q, /, =, W. See Section 9 for possible Reader settings.
>	Causes the Reader to send a string of text to the Host as a *z* packet; *data* contains the text to send. (The Reader will respond with a *z* packet containing the text.)
@	Causes the Reader to reset its internal date/timestamp to the specified time; *data* contains the date and/or time in one of the following formats. yyyy-mm-dd hh:mm:ss yyyy-mm-dd hh:mm hh:mm:ss hh:mm **Note:** the separators are optional; only digits are significant. (The Reader will respond with *d* or *e*.) Examples: Set to midnight: @00:00 Set to Sept 1, 2005 11:52:02 PM: @2005-09-01 23:52:02
A	Notifies the Reader that the previously sent data were rejected for one of the following reasons: • The packet was encrypted and the decryption failed. • The Host (CodeXML® Modem) is locked to a different Reader. The Reader should indicate to the user that the packet has been rejected; e.g., it may sound error beeps. See related register 12F: notify-of-packet-rejection. (The Reader will not respond to the Host.)
C	Apply a value to a register on the reader and save the value; data is in one of the following formats: C(XXX)YYY where XXX is the register number and YYY is the setting value, both in ASCII hex. This will change the value in the register and save it. For example C(26)64 will change the value of register 26 to 0x64 and save this setting. C/(XXX)YYY where XXX is the register number and YYY is the setting value, both in ASCII hex. This will toggle the bits in YYY and then save the resulting value (see / command). For example C/(1F7)40 will toggle bit 6 of register 1F7 and save this setting. CO(XXX)YYY where XXX is the register number and YYY is the setting value, both in ASCII hex. This will set the bits in YYY (change those bits to 1s) for register XXX (see O command). For example CO(1F7)40 will set bit 6 of register 1F7 high and save this setting. CQ(XXX)YYY where XXX is the register number and YYY is the setting value, both in ASCII hex. This will clear the bits in YYY (change those bits to 0s) for register XXX (see Q command). For example CQ(1F7)40 will clear bit 6 of register 1F7 and save this setting.

<table>
<tr><td></td><td><b>Note:</b> The C command can be used to set prefixes and suffixes. Please refer to section 7.4 for more information.
<b>Note:</b> The C command saves changes immediately and those settings will survive a reboot. It is equivalent to issuing a P and =, then a W command. C(2B)0 is the same as these three commands combined: P(2B)0, =(2B)0, W
(The Reader will respond with d or e.)
<b>Note</b>: Also see commands O, P, Q, /, =, W. See Section 9 for possible Reader settings.</td></tr>
<tr><td>G</td><td>Get setting from Reader if followed by a register number in parenthesis.
G([prefix|suffix]) will return the prefix or suffix text.
(The Reader will respond with d and the setting value or with e. The setting value, if numeric, will be in hexadecimal, padded to 8 digits.)
<b>Note:</b> see Section 9 for possible Reader settings.</td></tr>
<tr><td>H</td><td>Requires subcommand. See below.
$ – Lock
<table>
<tr><td>$</td><td>Commands to lock and unlock the reader with a PIN.
Lock and Unlock the reader using the CR5000AV Configurator section of the Web Configuration Guide.
By default, commands can be sent to the reader from a host (such as CortexTools or CortexMobile) even if the reader is locked. Commands can also be sent to a locked reader by embedding them in a Keyed Configuration Code. These Keyed Configuration Codes are generated by the Web Configuration Guide for CR5000AV commands, and will soon be generated by CortexTools and CortexMoble for general commands.
<b>Sub-Commands:</b>
<table>
<tr><th></th><th>Command</th><th>Description</th></tr>
<tr><td>R</td><td>Reset PIN and Unlock Reader</td><td>• If this command is sent to the reader via barcode the read must be triggered by a physical button press. The reader can not be in Continuous Scan or Motion Detection modes to read the barcode.
• The reader must be rebooted within 30 seconds of receiving this command in order to actually reset the reader. If the reader is not rebooted within 30 seconds, the reader will error beep and return to its previous, locked state.
<b>Example:</b> H$R</td></tr>
</table>
<b>Keyword:</b> #Lock</td></tr>
</table>
5 – CR5000 & CR5000RTC on-counter (In-Stand) or off-counter (Out-of-Stand) modes</td></tr>
</table>

5 Configures in on-counter (In-Stand) or off-counter (Out-of-Stand) behavior of the CR5000 & CR5000RTC to the specified mode. These codes are designed to configure complex modes in a single command, even if Code adds or changes some of the underlying commands at a later date.

Format: H5(in|out)<mode>

- in|out
  - in – configures the reader's on-counter (In-Stand) behavior
  - out – configures the reader's off-counter (Out-Of-Stand) behavior
- mode

#	Mode	Description
0	Trigger	Scan on trigger press, use targeting LEDs otherwise. Does not block duplicate barcodes.
1	Pick List	Scan on trigger press, use targeting LEDs otherwise. The barcode must be near the center of the reader's field of view to be decoded. Does not block duplicate barcodes.
2	Detect Barcode	Continuous scanning with IR illumination. If a barcode is detected, switch to red illumination. Will not output the barcode a second time while it remains in the field of view.
3	Detect Motion	Continuous scanning with IR illumination. If motion is detected, switch to red illumination. Will not output the barcode a second time while it remains in the field of view.
4	Always On	Continuous scanning with dim red illumination. If a barcode is detected, switch to normal red illumination. Will not output the barcode a second time while it remains in the field of view.

For example, to configure the reader for Trigger mode when handheld, use the H5(out)1 command. To configure the reader for Always On mode when on the counter, use the H5(in)4 command.

Keywords: #Stand

A – Encryption Commands

A

Defines commands that are used with the wireless packet data encryption feature (256 Bit AES). Packet data encryption between a reader and modem can only occur on the M2 modem with firmware 0187+. When this feature is enabled, it will encrypt the data portion of the reader packet sent to the modem. Please reference the Code knowledge base (search “AES Setup” at http://www.codecorp.com/knowledge-database.php) for detailed setup instructions.

Format: HA<sub-command>

Sub-Commands:

	Command	Description
e	Enable Encryption Feature	Enables Packet data encryption. Command will fail if reader is not connected to a modem. If the operation is successful, BOTH the reader and modem will blink the same pattern (on the wireless indicator for the reader). The pattern is two short blinks, one long, one short and a three second pause. A short blink is approximately 0.5 seconds and a long blink is 1 second. **Example:** HAe
d	Disable Encryption Feature	Disables packet data encryption. In order for modem encryption mode to be disabled, reader needs to be connected to the modem when this command is issued. **Example:** HAd

Keywords: #AES

T – T500 Commands

T

Defines commands sent from the Reader to the T500 Bluetooth® Cable. There must be a T500 connected to a reader in order for these commands to have any effect. These commands induce the T500 to perform actions.

Format: HT<sub-command>

Sub-Commands:

	Command	Description
c	Connect	Passes QuickConnect Code™ data from the host reader to the T500, which in turn attempts to connect to the specified Bluetooth® address. **Format:** HTc<Bluetooth®_Address> Where <Bluetooth®_Address> is the target Bluetooth® address represented in 12 hexadecimal characters. **Example:** HTcAC220B48F040

<table>
<tr><td></td><td></td><td>d</td><td>Forget Paired Bluetooth® Devices</td><td>This command will cause the T500 to “forget” all connection information associated with prior Bluetooth® hosts.   After sending this command, it will require re-pairing for the devices to connect again.</td></tr>
<tr><td></td><td></td><td>i</td><td>Information</td><td>Information about the T500, forwarded to the wirelessly connected host. Broken further into sub-commands.
Format: HTi<sub-command>
<table>
<tr><th></th><th>Command</th><th>Description</th></tr>
<tr><td>b</td><td>Bluetooth®</td><td>Sends Bluetooth® address from T500 to wirelessly connected host.</td></tr>
<tr><td>s</td><td>Serial Number</td><td>A command sent to the T500 to have the T500 send its serial number to the wirelessly connected host.</td></tr>
<tr><td>f</td><td>Friendly Name</td><td>Sends Bluetooth® friendly name to wirelessly connected host.</td></tr>
</table></td></tr>
<tr><td></td><td></td><td>f</td><td>Set Friendly Name</td><td>Friendly name follows the ‘f’, up to 30 characters. If more than 30 characters, the name will be truncated.
Example: HTfThis is a new name</td></tr>
<tr><td></td><td></td><td>k</td><td>Terminate Connection</td><td>This command will terminate the active Bluetooth® connection and clear the “last connected” index so that the T500 will not auto-reconnect to the last host</td></tr>
<tr><td></td><td></td><td>s</td><td>Change setting</td><td>These command will change settings or behavior of the T500 as described in the following table:
Format: HTs<sub-command><Setting Value>
To enter “Setting Value” in decimal, prefix the number with ‘#’, otherwise the firmware will interpret the value as a hexadecimal number.
Example: To set the auto-reconnect timer to 20 seconds, use one of the following commands:
HTsr#20000 **OR** HTsr4E20
<table>
<tr><th></th><th>Command</th><th>Default (Hex)</th><th>Description</th></tr>
<tr><td>r</td><td>Auto-reconnect Timer</td><td>7530
(#30,000)</td><td>**Valid Range:** 3E8 (#1000) to FFFFFFFF Milliseconds
When disconnected, the T500 will attempt to reconnect to the last host periodically as determined by this setting value (in ms).</td></tr>
</table></td></tr>
</table>

<table>
<tr><td></td><td>
<table>
<tr><td></td><td></td><td></td><td>Values less than 3E8 (#1000) are unsupported.
To disable the auto-reconnect feature, set to 0xFFFFFFFF (#-1).</td></tr>
<tr><td>p</td><td>Pre-emptible Mode</td><td>0</td><td>This setting determines whether a remote Bluetooth® host may disconnect an existing host already connected to the T500 and then connect to that T500.
0 – Bluetooth® Hosts will not be able to connect when the T500 is already connected to a different host.
1 – Only Bluetooth® hosts that have previously paired with the T500 will be able to connect to the T500 when it has an existing connection. It will not be discoverable to non-paired hosts.
2 – All hosts may connect to the T500 when it is currently connected to another host.</td></tr>
</table>
<b>Keywords:</b> #T500
All other sub commands are reserved.</td></tr>
<tr><td>I</td><td>Requests the Reader to send its information string (no <i>data</i>).
(The Reader will respond with <i>i</i> or <i>e</i>.)
Optional Subcommands:
<table>
<tr><td>Is</td><td>Retrieves Settings data
Is[scd]
s – saved settings (same as ‘<’ command)
c – current settings (same as ‘,’ command
d – default settings</td></tr>
<tr><td>Ib</td><td>Sends XML formatted battery information to Host.</td></tr>
</table></td></tr>
<tr><td>J</td><td>Requests the Reader to restore settings to defaults (no <i>data</i>).
<b>Note:</b> J(prefix) and J(suffix) set the respective prefix or suffix to their default value. For most readers this is clearing the prefix or suffix. For the CR900FD, however, the suffix is reset to a CodeXML® encoded enter key.
(The Reader will respond with <i>d</i> or <i>e</i>.)</td></tr>
<tr><td>J1</td><td>Complete restore of factory setup. Will overwrite the apps and settings.</td></tr>
</table>

L	Requests the Reader to send a list of its stored files *data* is: • (no *data*) or “0”; all non-hidden files. • “1”; hidden and non-hidden files (The Reader will respond in the same manner as with the ‘(’ command, each *z* packet containing a file name as a NUL-terminated string of printable ASCII characters.)
N	Deletes all stored images (.jpg and .pgm) and buffered scan data (.log and .buf files)
O	Set a bit (or bits) in a Reader setting; *data* contains a printable ASCII string in the following format: hexadecimal register number in parentheses followed by a 32-bit signed integer value, expressed in ASCII hexadecimal characters (with optional minus sign) or ASCII decimal characters preceded by the ‘#’ character, e.g., O(2e)1000 or O(2e)#4096. The specified integer is ORed with the existing setting value. The way an O command is handled is equivalent to a P command – the effects are immediate but won’t survive a reboot. If you want the setting to be set after a reboot, issue an O then W or use the newer CO combination. (The Reader will respond with *d* or *e*.) **Note:** see Section 9 for possible Reader settings.
P	Put a value in a Reader register; *data* contains a printable ASCII string in the following format: hexadecimal register number in parentheses followed by a 32-bit signed integer value, expressed in ASCII hexadecimal characters (with optional minus sign) or ASCII decimal characters preceded by the ‘#’ character, e.g., /(2e)1000 or /(2e)#4096; the specified integer replaces the existing register value. (The Reader will respond with *d* or *e*.) **Note:** The P command can be used to set prefixes and suffixes. Please refer to section 7.4 for more information. **Note:** The P command saves changes immediately but those settings will *not* survive a reboot. If you issue a P(2B)0 command to change the value of register 2B from 1, then reboot the reader, the value of the 2B register will be 1 when the reader is ready again. In order to have that setting survive a reboot you must either issue a P then a W command or use the newer C command. C(2B)0 is the same as these three commands combined: P(2B)0, W (The Reader will respond with d or e.) **Note**: Also see commands O, P, Q, /, =, W. See Section 9 for possible Reader settings.
Q	Clear a bit (or bits) in a Reader register; *data* contains a printable ASCII string in the following format: hexadecimal register number in parentheses followed by a 32-bit signed integer value, expressed in ASCII hexadecimal characters (with optional minus sign) or ASCII decimal characters preceded by the ‘#’ character, e.g., Q(2e)1000 or Q(2e)#4096. The ones-complement of the specified integer is AND’ed with the existing register value.

	The way a Q command is handled is equivalent to a P command – the effects are immediate but won't survive a reboot. If you want the setting to be set after a reboot, issue a Q then W or use the newer CQ combination. (The Reader will respond with *d* or *e*.) **Note**: Also see commands O, P, Q, /, =, W. See Section 9 for possible Reader settings.
R	Requests that the previously sent packet be re-sent by the Reader; *data* may specify a maximum packet size the receiver will accept: *data* is either empty or specifies a 16-bit big-endian unsigned integer (2 bytes). If *data* is empty or specifies a size less than 32 (the minimum packet size), the Reader will use its preferred maximum packet size. Otherwise, it will use the specified max packet size (or less) and will fragment data across multiple smaller packets when necessary. (The Reader will respond by resending its previous packet or with *e* if there was no previous packet. If the max data size has changed, it may resend the previous data in a sequence of more than one packet.)
T	Requests the current date and time (no *data*) (The Reader will respond with *d* with *data* containing the date and time formatted as yyyy-mm-dd hh:mm:ss.)
W	Requests the Reader to write its current settings from RAM to its non-volatile memory. (The Reader will respond with *d* or *e*.) **Note:** The W command saves changes that have been make with a /, O, P or Q so those settings will survive a reboot. If you issue a P(2B)0 command to change the value of register 2B from 1, then reboot the reader, the value of the 2B register will be 1 when the reader is ready again. In order to have that setting survive a reboot you must either issue a P then a W command or use the newer C command. C(2B)0 is the same as these two commands combined: P(2B)0, W (The Reader will respond with d or e.) **Note**: Also see commands O, P, Q, /, =, W. See Section 9 for possible Reader settings.
Y	Acknowledge the receipt of a packet; *data* specifies the received packet number (one byte). (The Reader will not respond.)
Z	Request the Reader to reboot *data* is: • *empty* or '0'; reboot the Reader. • '1'; restart application. (The Reader will respond with *d* or *e* before it reboots.)
^	Requests the Reader to upload the specified stored file; *data* contains the file name, terminated with ASCII NUL. The Reader will respond with: • A 'g' packet containing "*filename<tab>*(*size*)"

	• One or more ‘z’ packet(s) • A ‘d’ packet containing “EOF<*tab*>(*CRC16*)” **Note:** *filename* “help” is reserved to send command information.
_	Causes the Reader to wait for all buttons to be released and clear its event queue (The Reader will respond with *d* or *e*.)
|	Process *data* as a decoded string. (The Reader will respond with *d* or *e*.)

7.4 Prefix and Suffix Handling

Prefixes and Suffixes are defined in text files (.prefix and .suffix) that reside on the reader. Readers with default prefixes and suffixes are defined in the files .prefix.default

C([‘prefix’|’suffix’])<URL encoded *text*> will set the prefix or suffix to *text*. The prefix and suffix each have a maximum length of 64 bytes. The URL encoded text must be in UTF-8 format. For standard ASCII characters, this encoding is the same as ASCII. For extended ASCII (values of 0x80 through 0xFF) the characters must be encode as a 2-byte sequence starting with either 0xC2 or 0xC3 (see http://www.fileformat.info/info/charset/UTF-8/list.htm or http://en.wikipedia.org/wiki/UTF-8). Exception: if JavaScript is being bypassed during decoding, then the prefix/suffix may be in extended ASCII.

Example: C(suffix)%01X%1ean//n%04 will create a CodeXML® sequence to send the scancode for the enter key.

Example: C(prefix)ol%c3%a9 will create a prefix of “olé”

8 File Installation

8.1 Simple Protocol

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

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

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

9 Reader Settings

The Host sets the Reader settings using the */, C, O*, *P*, *Q,* and = commands and reads them using the *G, ‘,’,* and < commands.

For example, the following C command sets register 2E to the value 0x7F.

```
C(2E)7F
```

Note: for two-digit register numbers (i.e. settings 00 through FD), an alternative format may be used: in place of the parentheses and hexadecimal setting number, substitute a single character, which represents the setting number. The equivalent to the example above is

```
C.7F
```

The ASCII ‘.’ character has the hexadecimal value 0x2E. In certain circumstances, such as with text-commands, “percent-encoding” may be used for encoding a character as a sequence consisting of the percent character followed by two hexadecimal digits. With percent-encoding, the example may be expressed as

```
C%2E7F
```

In Section 9.3 below, the **Reg** column is the register number, in hexadecimal**,** to be used with the commands identified above. In the **Default** column, all values are in hexadecimal unless otherwise specified. To use decimal values in commands you must precede the data with a pound sign ‘#’. The following C command sets register 2E to the same value as the example above:

```
C(2E)#127
```

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

9.1 Binary Dip Switch

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

An example of this is register 0A, ‘NEC 2 of 5 Options’. The following settings are possible:

Bit (R to L)	Controls	Value
0	NEC 2 of 5 Decoding	0: Disabled
		1: Enabled
1	Enable checksum checking	0: Disabled
		1: Enabled

2	Remove checksum from the result (Checksum checking must be enabled for this to take effect)	0: Disabled 1: Enabled

Given the settings above, the binary string to turn on NEC 2 of 5 decoding with checksum checking and the checksum removed from the result string, is 111 (bits left to right). The same string would be 0x7 or decimal 7.

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

C(0A)7

Or

C(0A)#7

9.2 Field of Interest

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

At a given focus distance, the FOI 0 field is designed to read small, low-mil barcodes while the FOI 1 field is designed to pick up large, wide barcodes.

This document will refer to FOI 0 as HD and FOI 1 as Wide.

Note: The CR900FD is a single Field of Interest reader. There is no FOI 0 in the CR900FD.

9.3 Reader Setting Persistence

Some reader settings have special behaviors under certain conditions, such as when resetting to factory defaults. There are four categories for setting persistence, which are listed below.

Persistence	Description
Unprotected	**Unless otherwise specified, all settings have this type of persistence.** This setting can be changed using a command such as ‘P’ or saved using commands such as ‘C’ or ‘W’. If the reader is reset to factory defaults using a ‘J’ command, the setting will revert to its default value.
Protected	This setting can be changed or saved. If the reader is reset to its factory defaults using the ‘J’ command, the value of this setting will not change. However, if the settings file is deleted, the setting will revert to its default value on the next boot.

Preserved	This setting can be changed and saved. This setting is not affected by a 'J' command, nor by deleting the settings file.
Reset-on-boot	This setting can be changed but cannot be saved. On boot up the reader will revert to the default value. A 'J' command will also reset this value to default

9.4 Docked State Settings

Keywords: #Charger, #Stand

9.4.1 Overview

Some readers support the ability to change behavior, based on whether they are in a docked state. Docking can be achieved in two different ways. A reader is considered docked when it is placed in a stand that has hardware to indicate its presence, such as a charging base, or, in the case of the CR5000 & CR5000RTC, when it is placed on a surface. The undocked state is when the reader is removed from the stand or is picked up off a surface (again, for the CR5000 & CR5000RTC). These behavior changes are controlled in three different ways.

1. Wired readers (except the CR5000 & CR5000RTC) have a reed switch that is acted on by a magnet in the stand to indicate docked state. These readers check and act on the docked state in order to change behavior.
2. By default, wireless readers have an In-Charger Event register and an Out-of-Charger Event register. When the dock state of the reader changes, the corresponding event from the register will be fired. By default, the events enable battery charge level indication when docked.
3. The CR5000 & CR5000RTC utilizes a set of Standard registers that have corresponding In-Stand and Out-of-Stand registers for each. The In-Stand or Out-of-Stand register values are copied to the Standard registers when the reader detects a change in the docked state. Wireless readers can be also configured (using register 2AB) to leverage this In-Stand/Out-of-Stand/Standard register method. The In-Stand behavior takes effect when the wireless reader is placed in its charger base, and the Out-of-Stand behavior takes effect when the wireless reader is removed from its charger base.

For the docked/undocked settings to have any effect, the reader must be configured to detect a stand (bit 6 of register 1F7). For the Bluetooth®-enabled readers, the reader must also be configured to detect the charging base as a stand (register 2AB).

Changing the In-Stand and Out-of-Stand registers does not do anything immediately. The reader does not look at the In-Stand Default Event register (2C2), for example, to decide what to do when it is docked. Rather, the reader is always looking at the Default Event register (C4), but the Default Event setting is changed when the reader is docked or undocked. For example, when the reader is docked, the value of the In-Stand Default Event register (2C2) is copied to the Default Event register (C4).

The trigger for copying the In-Stand (or Out-of-Stand) settings to the corresponding standard settings is an event that gets posted by the firmware when it detects a change in the docked state (again, the charger for the Bluetooth® readers and the counter/table for the CR5000 & CR5000RTC). The “update behavior based on the stand” event (event 0xF2 as used by register C4 and other registers that utilize events) is the generic event that will trigger this operation.

9.4.2 H5 commands (CR5000 & CR5000RTC only)

The CR5000 & CR5000RTC support several standard presentation and handheld modes, which can be specified using the H5 subcommands. These commands are described in the Command Types section of this document. Of note here, however, is that these commands will overwrite the In-Stand or Out-of-Stand registers with pre-defined values, and perform the equivalent of event 0xF2 mentioned above, so that the In-Stand or Out-of-Stand settings take effect immediately. To create a custom configuration barcode that uses an H5 command, but also modifies one of the In-Stand or Out-of-Stand registers, first issue the H5 command, then modify the In-Stand or Out-of-Stand register(s), and then issue event 0xF2 to force the settings to take effect immediately. Note that if the custom configuration barcode does not modify any of the In-Stand or Out-of-Stand settings directly, then event 0xF2 is not necessary.

As an example, to create a custom configuration code that sets the beep volume of the CR5000 & CR5000RTC to 30% at all times and configures the on-counter (In-Stand) Trigger mode with the exception that targeting lights are not illuminated, type these commands (with or without all the comments) into a text editor:

```
H5(in)0 ;This command sets the built-in on-counter (In-Stand) Trigger Mode.
C(2DB)0 ;This command sets the In-Stand Target Enable register to 0, where the
        H5(in)0 would have set it to 1.
$%F2    ;This command updates the standard registers from either the In-Stand or
        Out-of-Stand registers, depending on the current state.   For example,
        if $%F2 is issued while the reader is in the Out-of-Stand state, the
        reader will copy all of the Out-of-Stand registers to the corresponding
        standard registers. (In this example, register 2DB is the In-Stand value
        for standard register 0F).
C(26)1E ;This command sets the beep volume to 30% of full by setting the
        register to 0x1E (30 in decimal). Standard commands, such as this one,
        are not affected by the In-Stand or Out-of-Stand state so they apply at
        all times.
```

Save the file with the .crb extension, then either loaded to the reader directly or convert it into a Data Matrix configuration code readable by the CR5000 & CR5000RTC using a Code barcode generation tool such as CortexTools®.

9.5 Reader Settings Table

<table>
<tr><th>Reg</th><th>Setting Name</th><th>Default (Hex)</th><th>Comment</th></tr>
<tr><td>00</td><td>Bluetooth® Radio Out-of-range indicator</td><td>0</td><td>Binary Dip Switch
<table>
<tr><th>Bit</th><th>Controls</th><th>Value</th></tr>
<tr><td>0</td><td>Vibrate</td><td>0: Disabled
1: Enabled</td></tr>
<tr><td>1</td><td>Beep</td><td>0: Disabled
1: Enabled</td></tr>
</table>

<b>Note:</b> Readers with a vibration motor are the CR1400, CR1400XHD, CR1428, CR2600, CR2600XHD, CR3600, CR3600DPM and CR6000.
<b>Keywords:</b> #Beep, #Vibration</td></tr>
<tr><td>04</td><td>Continuous Illumination During Read</td><td>0
CR5000:
CR5000RTC:
1</td><td>0: Minimal Illumination (Illumination stops whenever not needed during the Read Cycle – this produces a little more flashing but reduces power consumption to a degree)
1: Leave Illumination On Until End of the Read Cycle
Leave illumination on during read</td></tr>
<tr><td>05</td><td>USB Declaration Wait State</td><td>0</td><td>0: Declare enumeration after receipt of set LED status report
1: Declare enumeration after receipt of get report descriptor command (used for some Windows® CE-based devices)
2: Declare enumeration after the receipt of set configuration command.
Special case for USB enumeration that doesn’t require Host keyboard response
<b>Persistence:</b> Protected</td></tr>
<tr><td>08</td><td>Reader Packet Format</td><td>1</td><td>1: Raw Mode
2: Packet Mode
6: Safe Upgrade Mode (Version 1)
This setting is used in conjunction with registers 1B and 42 to configure the communication mode between standard “one-way” and “two-way” modes.
For example, USB “two-way” native:
1B: 5 (USB Native)
08: 2 (packet mode)
42: 1 (expect response)
<b>Persistence:</b> Protected
<b>Keywords:</b> #Communications</td></tr>
</table>

<table>
<tr><th>Reg</th><th>Setting Name</th><th>Default (Hex)</th><th>Comment</th></tr>
<tr><td>0A</td><td>NEC 2 of 5 Symbology</td><td>0</td><td>Binary Dip Switch
<table>
<tr><th>Bit</th><th>Controls</th><th>Value</th></tr>
<tr><td rowspan="2">0</td><td rowspan="2">NEC 2 of 5 Decoding</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">1</td><td rowspan="2">Enable checksum checking</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">2</td><td rowspan="2">Remove checksum from the result</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
</table>
Checksum checking must be enabled by setting bit 1 in order for checksum removal from the output (setting bit 2) to have any effect.
<b>Keywords:</b> #2Of5</td></tr>
<tr><td>0B</td><td>Matrix 2 of 5 Symbology</td><td>0</td><td>Binary Dip Switch
<table>
<tr><th>Bit</th><th>Controls</th><th>Value</th></tr>
<tr><td rowspan="2">0</td><td rowspan="2">Matrix 2 of 5 Decoding</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">1</td><td rowspan="2">Enable checksum checking</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">2</td><td rowspan="2">Remove checksum from the result</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
</table>
Checksum checking must be enabled by setting bit 1 in order for checksum removal from the output (setting bit 2) to have any effect.
<b>Keywords:</b> #2Of5</td></tr>
<tr><td>0C</td><td>Telepen Symbology</td><td>0</td><td>0: Disabled
1: Enabled
<b>Keywords:</b> #Telepen</td></tr>
<tr><td>0F</td><td>Targeting Control</td><td>1
CR5000:
CR5000RTC:
0</td><td>0: Targeting Disabled
1: Targeting Enabled
<b>Keywords:</b> #StandModes</td></tr>
</table>

<table>
<tr><th>Reg</th><th>Setting Name</th><th>Default (Hex)</th><th>Comment</th></tr>
<tr><td>13</td><td>1D Barcode Aggressiveness</td><td>0</td><td>0: Most Aggressive
1: Less Aggressive for poorly printed 1D barcodes
2: Least Aggressive for poorly printed 1D barcodes
#11: Less Aggressive for 1D barcodes with low module size (1X near 1.0 pixel per module)
#12: Least Aggressive for 1D barcodes with low module size (1X near 1.0 pixel per module)</td></tr>
<tr><td>14</td><td>Image Transform</td><td>0</td><td>0: No Transform
1: Mirror – Enables decoding of mirrored Data Matrix, QR, Han Xin, and Aztec symbologies. Those symbologies must be enabled.
<b>Keywords:</b> #Aztec, #DataMatrix, #HanXin, #QR</td></tr>
<tr><td>16</td><td>Data Matrix Rectangular Symbology</td><td>1</td><td>0: Disabled
1: Enabled
<b>Note:</b> This register has no effect on a CR900FD without the proper 2D barcode decoding license.
<b>Keywords:</b> #DataMatrix</td></tr>
<tr><td>19</td><td>Data Matrix Symbology</td><td>3</td><td>Binary Dip Switch
<table>
<tr><th>Bit</th><th>Controls</th><th>Value</th></tr>
<tr><td rowspan="2">0</td><td rowspan="2">Data Matrix Decoding</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">1</td><td rowspan="2">Inverse Data Matrix Decoding</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">2</td><td rowspan="2">Mirror Data Matrix Decoding</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
</table>
<b>Note:</b> This register has no effect on a CR900FD without the proper 2D barcode decoding license.<b>Keywords:</b> #DataMatrix</td></tr>
<tr><td>1A</td><td>Straight 2 of 5 Symbology</td><td>0</td><td>0: Disabled
1: Enabled
Straight 2 of 5 (with 2 or 3 start/stop codes) Decoding.
<b>Note:</b> Straight 2 of 5 with 2 start/stop is also called IATA 2 of 5. Straight 2 of 5 with 3 start/stop is also called Industrial 2 of 5. CD12.3.0 supports Industrial 2 of 5. CD 13.1.4 supports IATA 2 of 5.
<b>Keywords:</b> #2Of5</td></tr>
</table>

Reg	Setting Name	Default (Hex)	Comment
1B	Communications Mode	2 CR2300: CR2600: CR2600XHD: CR3600: CR3600DPM: 5 CR4405: C (#12) USB decode board: 2 RS232 decode board: 1	1: RS232 Serial Mode 2: USB Keyboard Mode 3: USB IBM POS Mode 4: Bluetooth® Serial Port Profile (SPP) Mode 5: USB HID Native Mode 6: USB Virtual COM Mode 7: USB HID POS (Terminal 131) Mode A (#10): Bluetooth® Keyboard Mode B (#11): Sled Apple® iOS App Mode C (#12): Sled Apple® iOS Keyboard Mode D (#13): USB Pass Through Mode E (#14): USB CDC Mode 10 (#16): T500 Bluetooth® Apple® iOS App Mode 12 (#18): T500 Bluetooth® Apple® iOS Keyboard Mode 13 (#19): T500 Bluetooth® Serial Port Profile (SPP) Mode 14 (#20): T500 Bluetooth® HID Keyboard Mode This setting is used in conjunction with registers 08 and 42 to configure the communication mode between standard "one-way" and "two-way" modes. For example, USB "two-way" native: 1b: 5 (USB Native) 08: 2 (packet mode) 42: 1 (expect response) **Note: To switch from USB keyboard mode to Downloader mode (HID), the following must be completed within 1 second:** • first output report with num lock set and caps lock clear • second output report with num lock set and caps lock clear • third output report with caps lock set and num lock clear • fourth output report with caps lock set num lock clear • fifth output report with num lock set and caps lock clear • sixth output report with num lock set and caps lock clear On the last output report comm protocol is set to raw mode, comm expect response is false and comm mode is USB Downloader (HID) mode. **Persistence:** Protected **Keywords:** #Communications

Reg	Setting Name	Default (Hex)	Comment
1C	Serial Baud Rate	1C200 (#115200)	All standard baud rates from 300 baud to 115200 baud • 12C (#300) • 4B0 (#1200) • 960 (#2400) • 12C0 (#4800) • 2580 (#9600) • 4B00 (#19200) • 9600 (#38400) • E100 (#57600) • 1C200 (#115200) **Persistence:** Protected **Keywords:** #SerialComm
1D	Serial Stop Bits	1	1: Send 1 Stop Bit 2: Send 2 Stop Bits **Keywords:** #SerialComm
1E	Serial Data Bits	8	7: 7 Data Bits 8: 8 Data Bits **Keywords:** #SerialComm
1F	Serial Flow Control	0	0: Disabled 1: Hardware **Note:** Undefined behavior will result if this setting is enabled when Output Good Read on RTS (0x02F1) is enabled. **Keywords:** #SerialComm
22	Serial Parity	0	0: None 1: Odd 2: Even **Keywords:** #SerialComm
26	Beep Volume (percent)	64 (#100)	**Valid Range:** 0 to 64 (#100) Percent This is the current percentage of full volume potential. **Keywords:** #Beep
29	PDF417 Symbology	1	0: Disabled 1: Enabled **Note:** This register has no effect on a CR900FD without the proper 2D barcode decoding license. **Keywords:** #PDF417

<table>
<tr><th>Reg</th><th>Setting Name</th><th>Default (Hex)</th><th>Comment</th></tr>
<tr><td>2A</td><td>Micro PDF417 Symbology</td><td>0</td><td>0: Disabled
1: Enabled
<b>Note:</b> This register has no effect on a CR900FD without the proper 2D barcode decoding license.
<b>Keywords:</b> #PDF417</td></tr>
<tr><td>2B</td><td>QR Code Symbology</td><td>1</td><td>Binary Dip Switch
<table>
<tr><th>Bit</th><th>Controls</th><th>Value</th></tr>
<tr><td rowspan="2">0</td><td rowspan="2">QR Code Standard Decoding</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">1</td><td rowspan="2">QR Code Inverse Decoding</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">2</td><td rowspan="2">Micro QR Code Decoding</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">3</td><td rowspan="2">Unused</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">4</td><td rowspan="2">QR Code Mirror Decoding</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">5</td><td rowspan="2">Model 1 QR Code Decoding</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
</table>
<b>Note:</b> This register has no effect on a CR900FD without the proper 2D barcode decoding license.
Micro must have Standard enabled to be enabled and Mirror must have Standard or Micro enabled to be enabled. Inverse works independently of Standard QR.
CD supports Model 1 even though Model 1 is considered obsolete.
<b>Keywords:</b> #QR</td></tr>
</table>

Reg	Setting Name	Default (Hex)	Comment
2C	Active Mode Countdown Timer (ms) Bluetooth®: Extra Cabled Active Time Timer (ms)	2710 (#10000)	**Valid Range:** 0 to 7FFFFFFF Milliseconds. This state: Active (Highest power usage) Next state: Idle Counts down to the end of Active Mode. To disable this timer, set the register value to -1 as in the following example which disables and saves the timer setting with the 'C' command: C(2C)#-1 Active Mode is the time between the last user interaction with the Reader (button press, etc.) or firmware interaction (communications, etc.) and the end of the Active Mode Countdown Timer. There are many user and firmware events that will reset the timer; therefore it may seem that the timer is longer than the value set. Bluetooth® readers: This timer only applies if in the charging unit and will get added to register 32 to get the time to idle state from active mode. **Keywords:** #ReaderState
2D	Keyboard Maps	0 CR5000: CR5000RTC: 2	Set the Keyboard Map by reading a Keyboard Language Settings barcode from the http://codecorp.com/ConfigGuide/. The corresponding Keyboard XML file to a chosen Keyboard Language Settings barcode must exist on the reader in order for the Keyboard Language Settings barcode to work. The reader will default to the US English Keyboard Map if the keyboard.profile file reference is invalid. Use the L command or reference the File Browser tab in CortexTools® to see a list of installed Keyboard Maps on a reader. To utilize custom keyboard map files, set this register to 2 and write a keyboard.profile file to the reader referencing the desired Keyboard XML File. The handling of non-printable characters has been moved to register 2D3. Keyboard mapping is handled through Keyboard XML files that are provided by Code or customized by the user. Information

<table>
<tr><th>Reg</th><th>Setting Name</th><th>Default (Hex)</th><th>Comment</th></tr>
<tr><td></td><td></td><td></td><td>on the Keyboard XML format can be found in the Appendix: Custom Keyboard XML File Formatting.
The reader will use the legacy Code US English.xml keyboard map. If this register is set to any value other than 2, the firmware will write a keyboard.profile file containing the name of the keyboard map in corresponding to the legacy value and set this register value back to 2, according to the table below:
<table>
<tr><th>When 2D is set to</th><th>A keyboard.profile file will be written and reference this XML file</th><th>Register 2D3 will be set to</th></tr>
<tr><td>0</td><td>English.xml (The legacy KB map)</td><td>0</td></tr>
<tr><td>1</td><td>USInternational_Win.xml</td><td>0</td></tr>
<tr><td>2</td><td>No change</td><td>No change</td></tr>
<tr><td>3</td><td>USEnglish_Win.xml</td><td>3</td></tr>
<tr><td>4</td><td>French_Win.xml</td><td>0</td></tr>
<tr><td>5</td><td>German_Win.xml</td><td>0</td></tr>
<tr><td>6</td><td>Japanese_Win.xml</td><td>0</td></tr>
<tr><td>7</td><td>USEnglish_Win.xml</td><td>1</td></tr>
<tr><td>8</td><td>SwissGerman_Win.xml</td><td>2</td></tr>
<tr><td>9</td><td>BelgianFrench_Win.xml</td><td>0</td></tr>
<tr><td>A</td><td>UnitedKingdom_Win.xml</td><td>0</td></tr>
<tr><td>B</td><td>LatinAmerican_Win.xml</td><td>0</td></tr>
</table>
<b>Persistence:</b> Preserved
<b>Note:</b> See Appendix: Custom Keyboard XML File Formatting
<b>Keywords:</b> #KeyboardMap</td></tr>
</table>

Reg	Setting Name	Default (Hex)	Comment
32	Idle Mode Countdown Timer (ms) Bluetooth® readers: Active Mode Countdown Timer (ms)	64 (#100)	**Valid Range:** 0 to 7FFFFFFF Milliseconds. This state: Idle (Mid power usage) Next state: Standby Counts down to the end of Idle Mode. To disable this timer, set the register value to -1 as in the following example which disables and saves the timer setting with the 'C' command: C(32)#-1 Idle Mode is the time after the Active Mode Countdown Timer has expired and before the Idle Mode Countdown Timer has expired. There are many user and firmware events that will reset the timer or send the reader back into a higher-power mode; therefore it may seem that the timer is longer than the value set. Bluetooth® readers: This is the time from active mode to idle mode when the unit is not in the charger. If in the charger, this gets added to register 2C to get the time from active to idle. Bluetooth® readers this state: Active (High power usage) Bluetooth® readers next state: Idle **Keywords:** #ReaderState
33	Target Time Before Decode (ms)	0	**Valid Range:** 0 to 7FFFFFFF Milliseconds This is the time between a trigger event and when the reader will capture an image. It is useful for positioning the reader over a specific barcode among multiple barcodes.
34	Maximum Candidate Decodes Per Read	1	The Reader will process up to this number of barcodes per "read barcode" event. If there are more than this many barcodes in the field of view and within target tolerance, only the first ones will be decoded. For fastest performance with single barcodes, set to 1. Valid range is 1 to 10(#16). **Keywords:** #CompositeBarcodes
35	Button Stay-Down Time (ms)	0	**Valid Range:** 0 to 7FFFFFFF Milliseconds Keep processing the "read barcode" events for this amount of time (act as if the button stays down for this time) **Keywords:** #ButtonStayDownTime

Reg	Setting Name	Default (Hex)	Comment
37	Host Acknowledgement Time Limit (ms)	2BC (#700)	**Valid Range:** 0 to 7FFFFFFF Milliseconds After sending data to Host, the Reader waits up to this amount of time for the acknowledgement from Host before declaring failure. **Keywords:** #Communications
39	Trigger 1	3	The specified event is posted upon press of this button. For the CR2600, CR2600XHD, and CR2300, trigger 1 is the button towards the back on the top of the reader. For the CR3600, trigger 1 is the button to the left of the center navigation buttons. **Note:** Register C4 contains the list of valid events. **Keywords:** #DefaultEvent
3A	Trigger 2	3	For the CR2600, CR2600XHD, and CR2300, trigger 2 is the button towards the front on the top of the reader. For the CR3600, trigger 2 is the button to the right of the center navigation buttons. **Note:** Register C4 contains the list of valid events. **Keywords:** #DefaultEvent
3B	Trigger 3	0	The combination of pressing buttons defined by trigger 1 and trigger 2 being pressed simultaneously. **Note:** Register C4 contains the list of valid events. **Keywords:** #DefaultEvent
3C	Trigger 4	3	For the CR2600, CR2600XHD, CR2300, and CR3600, trigger 4 is the handle trigger. **Note:** Register C4 contains the list of valid events. **Keywords:** #DefaultEvent
3D	Trigger 5	0	The combination of pressing buttons defined by trigger 1 and trigger 4 being pressed simultaneously. **Note:** Register C4 contains the list of valid events. **Keywords:** #DefaultEvent
3E	Trigger 6	0	The combination of pressing buttons defined by trigger 2 and trigger 4 being pressed simultaneously. **Note:** Register C4 contains the list of valid events. **Keywords:** #DefaultEvent

Reg	Setting Name	Default (Hex)	Comment
3F	Trigger 7	0	The combination of pressing buttons defined by trigger 1, trigger 2 and trigger 4 being pressed simultaneously. **Note:** Register C4 contains the list of valid events. **Keywords:** #DefaultEvent
40	Text Command Timeout (ms)	2AF8 (#11000)	**Valid Range:** 0 to 7FFFFFFF Milliseconds The maximum time during which a complete text command from Host must be received. (Pending text command data is discarded when the timeout is exceeded.) **Keywords:** #Communications, #TextCommands

Reg	Setting Name	Default (Hex)	Comment
41	Text Commands	8	Binary Dip Switch

Bit	Controls	Value
0	Text Commands	0: Disabled
		1: Enabled
1	Suppress Echo	0: Disabled
		1: Enabled
2	Suppress Responses	0: Disabled
		1: Enabled
3	Disable Text Commands but Enable Magic Sequence; See Below	0: Disabled
		1: Enabled
4	Suppress URL Decode; See Below	0: Disabled
		1: Enabled
5	Accept On Timeout	0: Disabled
		1: Enabled

Magic Sequence:
The Magic Sequence is the string “;>PAx” where x is 1, 3, or 7 as defined above. This would normally be used in command text files, which would begin with the text-command-on sequence and end with the command to return to this special mode. For example:

```
;>PA7
;any desired commands here
PA8
```

Suppress URL Decode:
For example, if enabled, P%418 will not equal PA8.
The % is not recognized as an escape character

Accept On Timeout:
Note: See register 156 for details

Keywords: #TextCommands

<table>
<tr><th>Reg</th><th>Setting Name</th><th>Default (Hex)</th><th>Comment</th></tr>
<tr><td>42</td><td>Expect Acknowledgement From Host</td><td>0</td><td>0: Reader doesn’t wait for acknowledge
1: Reader will retransmit data when Host doesn’t acknowledge receipt

This setting is used in conjunction with registers 08 and 1B to configure the communication mode between standard “one-way” and “two-way” modes.

For example, USB “two-way” native:
1B: 5 (USB Native)
08: 2 (packet mode)
42: 1 (expect response)

<b>Persistence:</b> Protected

<b>Keywords:</b> #Communications</td></tr>
<tr><td>43</td><td>JPEG Picture Quality (percent)</td><td>32
(#50)</td><td><b>Valid Range:</b> 0 to 64 (#100) Percent

0: Raw Image (No JPEG Compression)
1 To 100: JPEG Compression Quality Percent

<b>Keywords:</b> #PictureSettings</td></tr>
<tr><td>46</td><td>Prefix With Timestamp</td><td>0</td><td></td></tr>
<tr><td>47</td><td>MaxiCode Symbology</td><td>0</td><td>Binary Dip Switch
<table>
<tr><th>Bit</th><th>Controls</th><th>Value</th></tr>
<tr><td rowspan="2">0</td><td rowspan="2">MaxiCode Decoding, Mode 0</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">1</td><td rowspan="2">MaxiCode Decoding, Mode 1</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">2</td><td rowspan="2">MaxiCode Decoding, Mode 2</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">3</td><td rowspan="2">MaxiCode Decoding, Mode 3</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">4</td><td rowspan="2">MaxiCode Decoding, Mode 4</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">5</td><td rowspan="2">MaxiCode Decoding, Mode 5</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">6</td><td rowspan="2">MaxiCode Decoding, Mode 6</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
</table>
<b>Note:</b> This register has no effect on a CR900FD without the proper 2D barcode decoding license.</td></tr>
</table>

Reg	Setting Name	Default (Hex)	Comment
48	Codabar Checksum	0	0: Disable checksum checking and output checksum character if one exists 1: Enable checksum checking and output checksum character 2: Enable checksum checking and remove checksum character from output If this register is set to a value greater than zero and no checksum is present in the barcode or the checksum is incorrect, the barcode will not read. **Keywords:** #Codabar
49	Code 39 Full ASCII Symbology	0	0: Disabled 1: Enabled Code 39 Full ASCII Decoding **Keywords:** #Code39
4A	Composite Barcodes	0	0: Disabled 1: Enabled Composite Barcode Decoding **Keywords:** #CompositeBarcodes

<table>
<tr><th>Reg</th><th>Setting Name</th><th>Default (Hex)</th><th>Comment</th></tr>
<tr><td>4B</td><td>Postal Barcode Symbology</td><td>0</td><td>Binary Dip Switch
<table>
<tr><th>Bit</th><th>Controls</th><th>Value</th></tr>
<tr><td rowspan="2">0</td><td rowspan="2">USPS Postnet</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">1</td><td rowspan="2">USPS Planet</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">2</td><td rowspan="2">USPS Intelligent Mail</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">3</td><td rowspan="2">Australia Post</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">4</td><td rowspan="2">Dutch Post (KIX)</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">5</td><td rowspan="2">Japan Post</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">6</td><td rowspan="2">UK Royal Mail</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">7</td><td rowspan="2">UPU ID-tags</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
</table>
<b>Keywords:</b> #Postal</td></tr>
</table>

<table>
<tr><th>Reg</th><th>Setting Name</th><th>Default (Hex)</th><th>Comment</th></tr>
<tr><td>4C</td><td>GS1 DataBar™ Symbology</td><td>1F
(#31)</td><td>Binary Dip Switch
<table>
<tr><th>Bit</th><th>Controls</th><th>Value</th></tr>
<tr><td rowspan="2">0</td><td rowspan="2">GS1 DataBar™ Expanded decoding</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">1</td><td rowspan="2">GS1 DataBar™ Expanded Stacked decoding</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">2</td><td rowspan="2">GS1 DataBar™ Limited decoding</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">3</td><td rowspan="2">GS1 DataBar™ Omnidirectional and GS1 DataBar™ Truncated decoding</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">4</td><td rowspan="2">GS1 DataBar™ Stacked and GS1 DataBar™ Stacked Omnidirectional decoding</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
</table>
<b>Keywords:</b> #GS1Databar</td></tr>
<tr><td>4D</td><td>UPC Expansion</td><td>0</td><td>0: Disabled
1: Enabled
This register enables converting UPC-E output to UPC-A format
<b>Keywords:</b> #UPC</td></tr>
<tr><td>4E</td><td>UPC/EAN Supplemental</td><td>0</td><td>0: Disabled
1: Enabled
This register enables concatenating a standard UPC with the secondary supplemental UPC or EAN data (two or five digits), when present.
<b>Keywords:</b> #UPC, #EAN</td></tr>
</table>

<table>
<tr><th>Reg</th><th>Setting Name</th><th>Default (Hex)</th><th>Comment</th></tr>
<tr><td>4F</td><td>MSI Plessey Symbology</td><td>0</td><td>Binary Dip Switch with bits 1-3 comprising one value
<table>
<tr><th>Bit</th><th>Controls</th><th>Value</th></tr>
<tr><td rowspan="2">0</td><td rowspan="2">MSI Plessey</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td colspan="2" rowspan="8"><b>Bits 1-3</b>
This is a value based setting.
Calculate and compare checksums of this type and/or removal of the checksum (See examples below)</td><td>0: Disabled checksum checking</td></tr>
<tr><td>1: One Byte Modulus 10</td></tr>
<tr><td>2: One Byte Modulus 10 & One Byte Modulus 11</td></tr>
<tr><td>3: Two Bytes Modulus 10</td></tr>
<tr><td>4: reserved</td></tr>
<tr><td>5: One byte Modulus 10 then strip</td></tr>
<tr><td>6: check modulus 10 & 11 then strip</td></tr>
<tr><td>7: check two modulus 10 then strip</td></tr>
<tr><td rowspan="2">5</td><td rowspan="2">UK Plessey (original)</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
</table>
<b>Examples:</b> To decode MSI Plessey:
<ul>
<li>MSI Mod10: set 4F to 3 (hex)</li>
<li>MSI Mod11/10: set 4F to 5 (hex)</li>
<li>MSI Mod10/10: set 4F to 7 (hex)</li>
<li>MSI Mod10 with removal, set 4F to B (hex)</li>
<li>MSI Mod11/10 with removal: set 4F to D (hex)</li>
<li>MSI Mod10/10 with removal: set 4F to F (hex)</li>
</ul></td></tr>
</table>

<table>
<tr><th>Reg</th><th>Setting Name</th><th>Default (Hex)</th><th>Comment</th></tr>
<tr><td>50</td><td>Aztec Symbology</td><td>1</td><td>Binary Dip Switch
<table>
<tr><th>Bit</th><th>Controls</th><th>Value</th></tr>
<tr><td rowspan="2">0</td><td rowspan="2">Aztec Decoding</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">1</td><td rowspan="2">Inverse Aztec Decoding</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">2</td><td rowspan="2">Mirror Aztec Decoding</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
</table>
<b>Note:</b> This register has no effect on a CR900FD without the proper 2D barcode decoding license.
<b>Keywords:</b> #Aztec</td></tr>
<tr><td>53</td><td>Decoder HD field (FOI 0) Width</td><td>280
(#640)</td><td><b>Valid Range:</b> 1 to 640 pixels
Decoder uses only the specified pixel width in the HD field (FOI 0). See section 9.2
<b>Note:</b> The High Density field (FOI 0) does not exist on the CR900FD.</td></tr>
<tr><td>54</td><td>Decoder HD field (FOI 0) Height</td><td>3C0
(#960)</td><td><b>Valid Range:</b> 1 to 960 pixels
Decoder uses only the specified pixel height in the HD field (FOI 0). See section 9.2
<b>Note:</b> The High Density field (FOI 0) does not exist on the CR900FD.</td></tr>
<tr><td>55</td><td>Notify Of Read Failure</td><td>0</td><td>0: Disabled
1: Send “r” packet on no-read (See “r” packet in Section 6.2)
0x100xx: post event on no-read, where the lower 8 bits specify the event number. For example, 0x10009 to post Event 0x09.
<b>Note:</b> Register C4 contains the list of valid events.
<b>Keywords:</b> #DefaultEvent</td></tr>
<tr><td>59</td><td>Beep Duration</td><td>64
(#100)</td><td><b>Valid Range:</b> 0 to 7FFFFFFF Milliseconds
Affects beep and vibrate duration.
<b>Keywords:</b> #Beep</td></tr>
</table>

<table>
<tr><th>Reg</th><th>Setting Name</th><th>Default (Hex)</th><th>Comment</th></tr>
<tr><td>6A</td><td>UPC/EAN Symbology</td><td>1</td><td>0: Disabled
1: Enabled
<b>Note:</b> Enables and disables all UPC and EAN barcodes (UPC-A, UPC-E, EAN-13, EAN-8)
<b>Keywords:</b> #UPC, #EAN</td></tr>
<tr><td>6B</td><td>Code 39 Symbology</td><td>1</td><td>Binary Dip Switch
<table>
<tr><th>Bit</th><th>Controls</th><th>Value</th></tr>
<tr><td rowspan="2">0</td><td rowspan="2">Enable/Disable Code 39</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">1</td><td rowspan="2">Enable/Disable Code 32</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
</table>
Code 32 is also known as Italian Pharmacode. It uses the same encoding as Code 39 and can be decoded as Code 39 with unintended data when Code 32 is not enabled. It is not necessary to enable Code 39 to enable Code 32.
<b>Keywords:</b> #Code39, #PharmaCode</td></tr>
<tr><td>6C</td><td>Code 93 Symbology</td><td>1</td><td>0: Disabled
1: Enabled</td></tr>
<tr><td>6D</td><td>Code 128 Symbology</td><td>1</td><td>0: Disabled
1: Enabled</td></tr>
<tr><td>6E</td><td>Interleaved 2 Of 5 Symbology</td><td>1</td><td>0: Disabled
1: Enabled
<b>Keywords:</b> #2Of5, #Interleaved2Of5</td></tr>
<tr><td>6F</td><td>Codabar Symbology</td><td>1</td><td>0: Disabled
1: Enabled
<b>Keywords:</b> #Codabar</td></tr>
<tr><td>70</td><td>Code 39 Checksum</td><td>0</td><td>0: Disable checksum checking and output checksum character if one exists
1: Enable checksum checking and output checksum character
2: Enable checksum checking and remove checksum character from output
If this register is set to a value greater than zero and no checksum is present in the barcode or the checksum is incorrect, the barcode will not read.
<b>Keywords:</b> #Code39</td></tr>
</table>

Reg	Setting Name	Default (Hex)	Comment
71	Interleaved 2 Of 5 Checksum	0	0: Disable checksum checking and output checksum character if one exists 1: Enable checksum checking and output checksum character 2: Enable checksum checking and remove checksum character from output If this register is set to a value greater than zero and no checksum is present in the barcode or the checksum is incorrect, the barcode will not read. **Keywords:** #2Of5, #Interleaved2Of5
72	Auto Stored Data Erase (Auto Log Erase)	1	0: Disable 1: Enable **Note:** When Enabled, data and images are cleared from nonvolatile memory when they are successfully uploaded to the Host. (In “Log mode,” this is set to Disabled). **Note:** Only applies to readers using Bluetooth® communication modes. **Keywords:** #DataRetention
73	Auto Buffer Upload	1	0: Disable 1: Enable When Enabled, the Reader will automatically upload buffered data (i.e. storage that hasn’t been previously uploaded) whenever a connection is present. **Note:** Only applies to readers using Bluetooth® communication modes. **Keywords:** #DataRetention
74	UPC Short Margin	1	0: Disabled 1: Enabled **Keywords:** #UPC

<table>
<tr><th>Reg</th><th>Setting Name</th><th>Default (Hex)</th><th>Comment</th></tr>
<tr><td>76</td><td>Send And Store Mode</td><td>0</td><td>0: Normal mode (buffered send)
1: Send and log mode
3: Log only mode
<b>Note:</b> Only applies to readers using Bluetooth® communication modes.
Also see register: B4. Reader will not allow the user to set auto-reconnect on, and store data. If any setting that allows storing data is turned on, auto-reconnect will be turned off (if it is on).
<b>Keywords:</b> #DataRetention</td></tr>
<tr><td>78</td><td>Settings Lock</td><td>1</td><td>1: Settings unlocked
3: Settings locked (except settings Lock)</td></tr>
<tr><td>85</td><td>Tri-Optic Options</td><td>0</td><td>Binary Dip Switch
<table>
<tr><th>Bit</th><th>Controls</th><th>Value</th></tr>
<tr><td rowspan="2">0</td><td rowspan="2">Tri-Optic Decoding, Normal Quiet Zones</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">1</td><td rowspan="2">Allow Short Quiet Zones</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">2</td><td rowspan="2">No Quiet Zones Required</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">3</td><td rowspan="2">Reverse order of first and second halves</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
</table>
</td></tr>
<tr><td>86</td><td>Motion Detection: Event</td><td>3</td><td><b>Valid Range:</b> Any event as defined in register 39 and re-listed in C4
Motion detection is enabled by setting register C4 to 0xF0.
This register is reset to 0 (disabled) when register C4 is changed away from 0xF0. When motion is detected, this event is posted.
See register 39 or C4 for list of events.
<b>Keywords:</b> #MotionDetection</td></tr>
</table>

Reg	Setting Name	Default (Hex)	Comment
88	Sleep Timeout (sec)	CR2300: CR2600: CR2600XHD: CR3600: CR3600DPM: A8C0 (#43200, 12 hours)	**Valid Range:** 0 to 7FFFFFFF Milliseconds. This state: Sleep (Lowest power usage) Next state: Power off Counts down to the end of Sleep Mode and then the device powers off. To disable this timer, set the register value to -1 as in the following example which disables and saves the timer setting with the 'C' command: C(88)#-1 **Note:** Only applies to readers using Bluetooth® communication modes. **Keywords:** #ReaderState
8C	USB Virtual COM Common Mode	1	0: Disabled 1: Enabled This and every other reader in USB VCOM Mode with USB Virtual COM Common Mode enabled will use a common VCOM port on the host PC. Any reader in USB VCOM Mode with this setting disabled will use a unique VCOM port on the host PC. **Persistence:** Protected **Note:** This setting is only valid for CR8000/1000/1400/900FD/6000 and can be enabled only with PC VCOM Driver 2.1.0 or above. **Keywords:** #Communications
8E	Time In Idle Mode (ms)	CR2300: CR2600: CR2600XHD: CR3600: CR3600DPM: 1388 (#5000)	**Valid Range:** 0 to 7FFFFFFF Milliseconds This register defines the amount of time in Idle Mode before entering Standby Mode. Other than the quicker transition to Active Mode, the Reader is in a state very similar to Standby Mode. Communications: Enabled Current state: Idle (Mid power usage) Next state: Standby **Note:** Only applies to readers using Bluetooth® communication modes. **Keywords:** #ReaderState

Reg	Setting Name	Default (Hex)	Comment
93	Suppress Beep On Decode	0	0: Beep indicating decode before JavaScript processing 1: Call JavaScript without beeping to indicate decode Normally, the Reader beeps as soon as decodes are read and processes them via JavaScript if necessary *after* the beep. To enable JavaScript to control the beep feedback, change this setting to 1; this will suppress the beep; the JavaScript would typically beep if the decode is valid or start another read cycle if it isn't. This setting does not suppress beeps for anything but a successful decode event. **Keywords:** #Beep
98	Decoder HD field (FOI 0) X Offset	0	**Valid Range:** 0 to 639 pixels Decoder uses the pixels after the specified pixel offset in the HD field (FOI 0). See section 9.2 **Note:** The High Density field (FOI 0) does not exist on the CR900FD.
99	Decoder HD field (FOI 0) Y Offset	0	**Valid Range:** 0 to 959 pixels Decoder uses the pixels after the specified pixel offset in the HD field (FOI 0). See section 9.2 **Note:** The High Density field (FOI 0) does not exist on the CR900FD.
9A	Decoder Wide field (FOI 1) X Offset	0	**Valid Range:** 0 to 639 pixels Decoder uses the pixels after the specified pixel offset in the Wide field (FOI 1). See section 9.2
9B	Decoder Wide field (FOI 1) Y Offset	0	**Valid Range:** 0 to 959 pixels Decoder uses the pixels after the specified pixel offset in the Wide field (FOI 1). See section 9.2

Reg	Setting Name	Default (Hex)	Comment
9D	Target Tolerance (percent)	640 (#1600)	**Valid Range:** 0 to 7FFFFFFF Percent For the Reader to accept a barcode, the barcode must be within a certain distance from the center of the image. The distance is defined as a percentage of the barcode's smaller dimension. For example, with a 10 x 20 mm barcode and a setting of 150 (%), the barcode must be within 15 mm of the center of the image. Any value over #1000 is considered infinite tolerance, and no target checking is performed. **Keywords:** #StandModes
9E	Extra Cabled Idle Time	CR2300: CR2600: CR2600XHD: CR3600: CR3600DPM: 2710 (#10000) CR4405: #-500	This is the time added to the idle timeout register 8E if unit is in the charger. To disable this timer, set the register value to -1 as in the following example which disables and saves the timer setting with the 'C' command: C(9E)#-1 Current State: Idle Next State: Standby **Note:** Only applies to readers using Bluetooth® communication modes. **Keywords:** #ReaderState

Reg	Setting Name	Default (Hex)	Comment
9F	Standby Mode Countdown Timer (ms)	FFFFFE0C (#-500) CR2300: CR2600: CR2600XHD: CR3600: CR3600DPM: 927C0 (#600000, 10 min) CR4405: EA60 (#60000, 1 min)	**Valid Range:** 0 to 7FFFFFFF Milliseconds. This state: Standby (Lowest power usage) Next state: Sleep (< 2mA current – NON-USB devices ONLY) Counts down to the end of Standby Mode. (Sleep mode NOT active in cabled USB devices). To disable this timer, set the register value to -1 as in the following example which disables and saves the timer setting with the 'C' command: C(9F)#-1 Standby Mode is the time after the Idle Mode Countdown Timer has expired and before the Standby Mode Countdown Timer has expired. There are many user and firmware events that will reset the timer or send the reader back into a higher-power mode; therefore it may seem that the timer is longer than the value set. **Note:** Only applies to readers using Bluetooth® communication modes. **Keywords:** #ReaderState
A1	Vibrate	1	0: Disabled 1: Enabled (vibration motor will be on at same time as speaker) **Note:** if vibrate-only is desired, set speaker volume to 0 using register 26. **Note:** Readers with a vibration motor are the CR1400, CR1400XHD, CR1428, CR2600, CR2600XHD, CR3600, CR3600DPM and CR6000.
A2	Default Event Delay (ms)	64 (#100) CR5000: CR5000RTC: 1	**Valid Range:** 0 to 7FFFFFFF Milliseconds The Reader will pause for this amount of time between each posting of the default event (used with "continuous read" mode). **Note:** Register C4 contains the list of valid events. **Keywords:** #DefaultEvent

Reg	Setting Name	Default (Hex)	Comment
A7	Beep (Vibrate) Pulse Separation (ms)	64 (#100)	**Valid Range:** 0 to 7FFFFFFF Milliseconds The spacing in milliseconds between beeps when beeping multiple times. **Note:** Readers with a vibration motor are the CR1400, CR1400XHD, CR1428, CR2600, CR2600XHD, CR3600, CR3600DPM and CR6000. **Keywords:** #Beep, #Vibration
AB	AGC Selection for Picture Taking	0	0: Use decoder AGC (designed for Symbology decoding) 1: Use imager AGC (optimized for pictures) **Keywords:** #PictureSettings
AC	Wide field (FOI 1) Picture Window Left Position	0	Specify left edge of window used with “take picture.” The position and size are relative to the virtual image (i.e. not the rotated physical image). **Note:** On a Code Reader, overall image is 960 pixels by 1280 pixels. Upper half is Wide field (FOI 1); lower half is High Density (FOI 0). The High Density field (FOI 0) does not exist on the CR900FD. **Keywords:** #PictureSettings
AD	Wide field (FOI 1) Picture Window Upper Position	0	Specify upper edge of window used with “take picture.” The position and size are relative to the virtual image (i.e. not the rotated physical image). **Note:** On a Code Reader, overall image is 960 pixels by 1280 pixels. Upper half is Wide field (FOI 1); lower half is High Density (FOI 0). The High Density field (FOI 0) does not exist on the CR900FD. **Keywords:** #PictureSettings
AE	Wide field (FOI 1) Picture Window Width	500 (#1280)	Specify width of window used with “take picture.” The position and size are relative to the virtual image (i.e. not the rotated physical image). **Note:** On a Code Reader, overall image is 960 pixels by 1280 pixels. Upper half is Wide field (FOI 1); lower half is High Density (FOI 0). The High Density field (FOI 0) does not exist on the CR900FD. **Keywords:** #PictureSettings

Reg	Setting Name	Default (Hex)	Comment
AF	Wide field (FOI 1) Picture Window Height	3C0 (#960)	Specify height of window used with “take picture.” The position and size are relative to the virtual image (i.e. not the rotated physical image). **Note:** On a Code Reader, overall image is 960 pixels by 1280 pixels. Upper half is Wide field (FOI 1); lower half is High Density (FOI 0). The High Density field (FOI 0) does not exist on the CR900FD. **Keywords:** #PictureSettings
B0	Target On Before Picture (ms)	3E8 (#1000)	**Valid Range:** 0 to 7FFFFFFF Milliseconds 0: Target off before picture capture **Keywords:** #PictureSettings
B3	Number Of Retries Before Reader Gives Up Sending Packet	3	**Valid Range:** 0 to FF Tries **Note:** The value 1 is defined as the original send attempt but no resends. The higher the number the longer each decode attempt can take. **Keywords:** #Communications
BF	USB Keyboard Poll Rate	A (#10)	**Valid Range:** 1 to FF (#255) Milliseconds The Host is requested to poll the USB device at the specified period.

Reg	Setting Name	Default (Hex)	Comment
C4	Default Event	FF (#255) CR900FD: CR1000: CR1000XHD: CR1400: CR1400XHD: CR1428: CR6000: CR8013: F0 (#240) CR5000: CR5000RTC: 6	The default value of FF is the idle event. This means that normally, no activity is occurring. However, if the value is set to one of those shown in the list below, then that action will be repeated continuously, unless interrupted by an external trigger or command. Any event other than 0 or FF will prevent the reader from entering power save mode. This register also causes register 86 to be set to 0, unless motion detection (F0) is selected as the default event. Event numbers (same as register 39) 0: No Action 1: Keep Awake 2: Show Target 3: Read In Both Fields (Default) 5: Read In High Density field (FOI 0). See section 9.2 **Note:** The CR900FD does not have a High Density field (FOI 0) 6: Read In Wide field (FOI 1). See section 9.2 7: Take Picture 8: Read In Most Recently Successful Field 0D to 50 (#13 To #80): Custom events (handled by JavaScript) DB (#219): Flash illumination and targeting DC (#220): Perform “Phone connected” handler DD (#221): Perform “Phone disconnected” handler DE (#222): Perform “Cable disconnected” handler DF (#223): Perform “Reader removed from charger” handler E0 (#224): Perform “Reader placed in charger” handler E1 (#225): Perform “Cable connected” handler E2 (#226): Beep F0 (#240): Detect motion F1 (#241): Toggle Apple® KB mode (Bluetooth® KB mode only) F2 (#242): Update behavior based on Stand Detection F4 (#244): Perform “Reader placed in stand” handler F5 (#245): Perform “Reader removed from stand” handler FF (#255): Idle **Keywords:** #Beep, #Charger, #DefaultEvent, #MotionDetection, #Stand, #StandModes

Reg	Setting Name	Default (Hex)	Comment
C6	Auto Connect Mode	1	0: No Auto Connect - connect only on “X” and “:” commands and upload events 1: Auto Connect - attempt to establish connection when in idle mode and maintain connection when in standby mode 2: Auto Reconnect - attempt to connect when there is data to send but only within specified time of last valid connection. 3: Auto Connect If Cabled - attempt to connect if Reader is cabled or in charger Bluetooth® readers: Also see register 76. Reader will not allow the user to set auto-reconnect on, and store data. If any setting that allows storing data is turned on, auto-reconnect will be turned off, if it is on. **Keywords:** #DataRetention
C7	Decoder Wide field (FOI 1) Width	280 (#640)	**Valid Range:** 1 to 280 (#640) pixels Decoder uses only the specified pixel height in the Wide field (FOI 1). See section 9.2
C8	Decoder Wide field (FOI 1) Height	3C0 (#960)	**Valid Range:** 1 to 3C0 (#960) pixels Decoder uses only the specified pixel height in the Wide field (FOI 1). See section 9.2

<table>
<tr><th>Reg</th><th>Setting Name</th><th>Default (Hex)</th><th>Comment</th></tr>
<tr><td>C9</td><td>Interleaved 2 Of 5 Lengths</td><td>0</td><td>
<b>Valid Range:</b> 0 to 64 (#100)
The number of digits allowed in an Interleaved 2 of 5 barcode, plus indicating support for a small quiet zone.
The length allowed is always even. If an Interleaved 2 of 5 has a smaller/shorter than standard quiet zone, 1 can be added to the length to tolerate the issue (but also increase the chance of partially decoding the barcode). Max value for C9 is 100 (100 digits without support for small quiet zones).
<b>Examples:</b>
To set length to 8, and
• To enable small quiet zone, set C9 to 9 (8 + 1)
• To disable small quiet zone, set C9 to 8 (8 + 0)
To set length to E (#14), and
• To enable small quiet zone, set C9 to F (#15) (E + 1)
• To disable small quiet zone, set C9 to E (#14) (E + 0)
<table>
<tr><th>Bit</th><th>Controls</th><th>Value</th></tr>
<tr><td rowspan="2">0</td><td rowspan="2">Small Quiet Zone</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td>[6:1]</td><td colspan="2">Number of Digits (Must be even and 64 (#100) or less)</td></tr>
<tr><td>[31:7]</td><td colspan="2">Reserved</td></tr>
</table>
<b>Keywords:</b> #2Of5, #Interleaved2Of5
</td></tr>
<tr><td>CA</td><td>Auto Disconnect</td><td>0</td><td>
0: Retain connection until explicit disconnect command
1: Disconnect from the Host when there is nothing to send. (In conjunction with registers C6 and 73, the Reader will connect when there is data to send, send the data, then disconnect (to allow another Reader to connect to the same Host).
2: Retain connection until reader enters sleep mode. Disconnect from the Host when the reader enters sleep mode.
<b>Note:</b> Only applies to readers using Bluetooth® communication modes.
</td></tr>
</table>

Reg	Setting Name	Default (Hex)	Comment
CE	Codablock F Symbology	0	0: Disabled 1: Enabled **Note:** This register has no effect on a CR900FD without the proper 2D barcode decoding license.
CF	Macro PDF417 Symbology	0	0: Disabled 1: Enabled **Note:** This register has no effect on a CR900FD without the proper 2D barcode decoding license. **Keywords:** #PDF417
D8	Composite Barcodes Require Both Elements	1	0: Accept any composite element 1: Only accept composite barcodes if both elements could be decoded. **Keywords:** #CompositeBarcodes
D9	Max Connection Wait Time	CR2300: CR2600: CR2600XHD: CR3600: CR3600DPM: 0F (#15) CR4405: 5	**Valid Range:** 0 to 7FFFFFFF Seconds The Reader will attempt connection for up to this amount of time when a connection is explicitly requested, such as when a QuickConnect Barcode is read or an upload is requested (by event or command). **Note:** Only applies to readers using Bluetooth® communication modes.
DB	Merge Selected Decodes	0	0: Disabled 1: Enabled When enabled, multiple valid decodes from a single read are concatenated. This includes Composite barcodes or when multiple barcode reading is enabled. **Keywords:** #CompositeBarcodes
E3	Trigger 1 Confirmation Time (ms)	0	**Valid Range:** 0 to 7FFFFFFF Milliseconds The trigger input must be asserted for this amount of time before the trigger event is generated. Setting this value > 0 makes it easier to select combinations of triggers (e.g. trigger 3, which is trigger 1 & 2 asserted together).
E4	Trigger 2 Confirmation Time (ms)	0	**Valid Range:** 0 to 7FFFFFFF Milliseconds The trigger input must be asserted for this amount of time before the trigger event is generated. Setting this value > 0 makes it easier to select combinations of triggers (e.g. trigger 3, which is trigger 1 & 2 asserted together).

Reg	Setting Name	Default (Hex)	Comment
E5	Trigger 3 Confirmation Time (ms)	0	**Valid Range:** 0 to 7FFFFFFF Milliseconds The trigger input must be asserted for this amount of time before the trigger event is generated. Setting this value > 0 makes it easier to select combinations of triggers (e.g. trigger 3, which is trigger 1 & 2 asserted together).
E6	Trigger 4 Confirmation Time (ms)	0	**Valid Range:** 0 to 7FFFFFFF Milliseconds The trigger input must be asserted for this amount of time before the trigger event is generated. Setting this value > 0 makes it easier to select combinations of triggers (e.g. trigger 3, which is trigger 1 & 2 asserted together).
E7	Trigger 5 Confirmation Time (ms)	0	**Valid Range:** 0 to 7FFFFFFF Milliseconds The trigger input must be asserted for this amount of time before the trigger event is generated. Setting this value > 0 makes it easier to select combinations of triggers (e.g. trigger 3, which is trigger 1 & 2 asserted together).
E8	Trigger 6 Confirmation Time (ms)	0	**Valid Range:** 0 to 7FFFFFFF Milliseconds The trigger input must be asserted for this amount of time before the trigger event is generated. Setting this value > 0 makes it easier to select combinations of triggers (e.g. trigger 3, which is trigger 1 & 2 asserted together).
E9	Trigger 7 Confirmation Time (ms)	0	**Valid Range:** 0 to 7FFFFFFF Milliseconds The trigger input must be asserted for this amount of time before the trigger event is generated. Setting this value > 0 makes it easier to select combinations of triggers (e.g. trigger 3, which is trigger 1 & 2 asserted together).
EB	Maximum Reader To Host Packet Data Size (bytes)	4000 (#16384)	**Valid Range:** 1 to 4000 (#16384) Bytes
EC	Host Acknowledgement Time Limit Multiplier (ms)	F (#15)	**Valid Range:** 0 to 7FFFFFFF Milliseconds When Expect Acknowledgement From Host (register 42) is nonzero, the Reader will wait up to Host Acknowledgement Time Limit (register 37) + dataSize * Host Acknowledgement Time Limit Multiplier (register EC) milliseconds to receive an acknowledgement from the Host.
ED	Prefix Decode Result With AIM Symbology Identifiers	0	0: Don't prefix with AIM identifier 1: Prefix decode result with ISO/IEC standard 15424/AIM symbology identifier **Keywords:** #AIMId

<table>
<tr><th>Reg</th><th>Setting Name</th><th>Default (Hex)</th><th>Comment</th></tr>
<tr><td>F0</td><td>Allow Code 128 Short Margin</td><td>1</td><td>0: Disabled
1: Enabled</td></tr>
<tr><td>F6</td><td>Code 39 Short Margin</td><td>1</td><td>0: disallow short margin Code 39 symbol decoding
1: allow short margin Code 39 symbol decoding
<b>Keywords:</b> #Code39</td></tr>
<tr><td>F7</td><td>Code 11 Symbology</td><td>0</td><td>Binary Dip Switch
<table>
<tr><th>Bit</th><th>Controls</th><th>Value</th></tr>
<tr><td>0</td><td>Code 11 Decoding with one or two checksum digits checked</td><td>0: Disabled</td></tr>
<tr><td></td><td></td><td>1: Enabled</td></tr>
<tr><td>2</td><td>Remove checksum(s) from the result string</td><td>0: Disabled</td></tr>
<tr><td></td><td></td><td>1: Enabled</td></tr>
</table>
<b>Note:</b> The firmware determines whether one-digit or two-digit checksum should be checked internally.</td></tr>
<tr><td>F8</td><td>PharmaCode Symbology</td><td>0</td><td>See Section <b>Error! Reference source not found.</b> - HYPERLINK \l "_PharmaCode" PharmaCode for more details.
Binary Dip Switch
<table>
<tr><th>Bit</th><th>Controls</th><th>Value</th></tr>
<tr><td>0</td><td>PharmaCode decoding</td><td>0: Disabled</td></tr>
<tr><td></td><td></td><td>1: Enabled</td></tr>
<tr><td>1</td><td>Support color bar</td><td>0: Disabled</td></tr>
<tr><td></td><td></td><td>1: Enabled</td></tr>
<tr><td>3</td><td>Reverse instead of Normal direction</td><td>0: Disabled</td></tr>
<tr><td></td><td></td><td>1: Enabled</td></tr>
</table>
<b>Note:</b> The reader decodes a PharmaCode in either horizontal or vertical orientation by default.
<b>Keywords:</b> #PharmaCode</td></tr>
<tr><td>F9</td><td>PharmaCode Bar Count</td><td>1004
(#4100)</td><td><b>Valid Range:</b> Each 8 bits can be 04 to 10 (#16)
Bit 0 – Bit 7: min bar count, 04 to 10 (#16)
Bit 9 – Bit 15: max bar count, 04 to 10 (#16)
<b>Keywords:</b> #PharmaCode</td></tr>
<tr><td>FA</td><td>PharmaCode Min Value</td><td>F
(#15)</td><td><b>Valid Range:</b> F (#15) to 1FFFE (#131070)
<b>Keywords:</b> #PharmaCode</td></tr>
<tr><td>FB</td><td>PharmaCode Max Value</td><td>1FFFE
(#131070)</td><td><b>Valid Range:</b> F (#15) to 1FFFE (#131070)
<b>Keywords:</b> #PharmaCode</td></tr>
</table>

<table>
<tr><th>Reg</th><th>Setting Name</th><th>Default (Hex)</th><th>Comment</th></tr>
<tr><td>FC</td><td>Keep reading barcodes as long as button is held down</td><td>0</td><td>0: Disabled (requires button to be released before next scan occurs)
1: Enabled
Note: When enabled, register 159 should be greater than zero.</td></tr>
<tr><td>10C</td><td>Bluetooth® Connected Cache Time (Seconds)</td><td>3</td><td>**Valid Range:** 0 to 7FFFFFFF Seconds
The time the last connection status received from the radio remains valid.
If a request is made during this time since last radio query, the cached status is returned. Otherwise, the Reader will query the radio for connection status (which takes up to 1 second).
Note: Only applies to readers using Bluetooth® communication modes.</td></tr>
<tr><td>10D</td><td>Grid Matrix Symbology</td><td>0</td><td>Binary Dip Switch<table><tr><th>Bit</th><th>Controls</th><th>Value</th></tr><tr><td>0</td><td>Grid Matrix</td><td>0: Disabled</td></tr><tr><td></td><td></td><td>1: Enabled</td></tr><tr><td></td><td>All other bits are reserved</td><td></td></tr></table></td></tr>
<tr><td>12D</td><td>Hong Kong 2 Of 5 Symbology</td><td>0</td><td>Binary Dip Switch<table><tr><th>Bit</th><th>Controls</th><th>Value</th></tr><tr><td>0</td><td>Hong Kong 2 of 5 Decoding</td><td>0: Disabled</td></tr><tr><td></td><td></td><td>1: Enabled</td></tr></table>**Keywords:** #2Of5</td></tr>
<tr><td>12F</td><td>Notify Of Packet Rejection</td><td>1</td><td>0: Disabled
1: Beep 3 times
0x100xx: post event on No-Read, where the lower 8 bits specify the event number. For example, 0x10009 to post Event 0x09.
Specify the behavior when a packet is rejected because of incorrect encryption key, incorrect packet protocol, or CodeXML® Modems locked to a different Reader.
Note: Register C4 contains the list of valid events.
Keywords: #Beep, #DefaultEvent</td></tr>
</table>

Reg	Setting Name	Default (Hex)	Comment
137	PDF417 Handle Invalid Shift	0	0: Disabled 1: Enabled Allows the decoding of PDF417 barcodes that were improperly encoded. **Note:** This register has no effect on a CR900FD without the proper 2D barcode decoding license. **Keywords:** #PDF417
154	Enable Black And White Pictures	0	0: Images remain in grey scale. 1: Captured images are converted to black and white. Converts grey scale images to black and white **Keywords:** #PictureSettings
159	Ignore Duplicate Barcode (ms)	0	**Valid Range:** 0 to 7FFFFFF Milliseconds Consecutive duplicate barcodes (i.e. barcodes that contain the same data) are blocked for this amount of time (in milliseconds). 0 turns off blocking of duplicate barcodes. **Note:** For the CR5000 & CR5000RTC only, the reader will continue to block duplicate barcodes as long as it continues to detect the barcode, and until the barcode has been removed for the “ignore duplicate barcode” time, if the time is not 0. In other words, for the CR5000 & CR5000RTC, to scan the same barcode twice, the barcode must be removed from the field of view for this period of time. **Keywords:** #StandModes, #DuplicateBlock
172	Automatically save the Bluetooth® connection address	CR2300: CR2600: CR2600XHD: CR3600: CR3600DPM: 1	0: Disable 1: Enable The address in the QuickConnect Barcode will be saved if enabled. **Note:** Only applies to readers using Bluetooth® communication modes.
173	Enable Bluetooth® Encryption	0	0: Disable 1: Enable **Note:** PIN must be set to 8 characters or more and the same value on the reader and remote device for connection to be successful. For CodeXML Modems, please refer to the D018582 Modem ICD document for details on the setPin command. **Keywords:** #Encryption

<table>
<tr><th>Reg</th><th>Setting Name</th><th>Default (Hex)</th><th>Comment</th></tr>
<tr><td>181</td><td>Glare Detection (Reduction)</td><td>0</td><td>0: Disabled
1: Enabled
Not available in the CR8000 because there is no off-axis illumination.</td></tr>
<tr><td>1EA</td><td>Good Read Duration (ms)</td><td>C8
(#200)</td><td><b>Valid Range:</b> 0 to 7FFFFFF Milliseconds
For the CR8000 and CR8013, this the duration that the good read hardware signal is asserted. For other readers, this is the duration for which the good read indicator LED will be illuminated and the pulse width of the good read signal on RTS.
<b>Keywords:</b> #GoodReadRTS</td></tr>
<tr><td>1F7</td><td>Platform configure buttons and Stand Detection</td><td>42
(#66)
CR2600:
CR2600XHD:
CR2300:
CR3600:
CR3600DPM:
2a
(#42)
CR4405:
2</td><td>Binary Dip Switch
<table>
<tr><th>Bit</th><th>Controls</th><th>Value</th></tr>
<tr><td rowspan="2">1</td><td rowspan="2">Button 1 Enable</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">3</td><td rowspan="2">Button2 Enable</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">5</td><td rowspan="2">Button4 Enable</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">6</td><td rowspan="2">Enable Stand Detection</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
</table>
<b>Note:</b> All other bits are Reserved.
To toggle Stand Detection, use CO(1F7)40 and CQ(1F7)40
<b>Keywords:</b> #StandDetect</td></tr>
<tr><td>20E</td><td>Motion Detection: Start Delay (ms)</td><td>0</td><td><b>Valid Range:</b> 0 to 7FFFFFFF Milliseconds
A built-in delay of 200 ms prevents motion detect from detecting motion right after a successful decode. This allows the barcode to be removed without triggering a new decode. Use this register to add an additional delay amount to the built-in delay.
<b>Keywords:</b> #MotionDetection</td></tr>
</table>

<table>
<tr><th>Reg</th><th>Setting Name</th><th>Default (Hex)</th><th>Comment</th></tr>
<tr><td>218</td><td>Motion Detection Option</td><td>0</td><td>Binary Dip Switch with bits 1 & 2 comprising a value
<table>
<tr><th>Bit</th><th>Controls</th><th>Value</th></tr>
<tr><td>1</td><td>CR6000 Illumination</td><td>0: Dark Field illumination
1: Diffuse Bright Field</td></tr>
<tr><td>2</td><td></td><td>2: Direct Bright Field
3: Dark Field and Diffuse Bright Field</td></tr>
</table>
<b>Keywords:</b> #DPM, #MotionDetection</td></tr>
<tr><td>21B</td><td>Reported Board Type</td><td>(See Comment column)</td><td><b>Valid Range:</b> 1 to 9999
The firmware will report this number as the “hardware type identifier” field of the reader information string.
<table>
<tr><th>Reader Type</th><th>Value</th></tr>
<tr><td>CR900FD:</td><td>384 (#900)</td></tr>
<tr><td>CR1000:</td><td>7</td></tr>
<tr><td>CR1000XHD:</td><td>7</td></tr>
<tr><td>CR1400:</td><td>8</td></tr>
<tr><td>CR1400XHD:</td><td>8</td></tr>
<tr><td>CR1428:</td><td>8</td></tr>
<tr><td>CR2300:</td><td>8FC (#2300)</td></tr>
<tr><td>CR2600</td><td>9</td></tr>
<tr><td>CR2600XHD:</td><td>9</td></tr>
<tr><td>CR3600:</td><td>A (#10)</td></tr>
<tr><td>CR3600DPM:</td><td>A (#10)</td></tr>
<tr><td>CR4100:</td><td>1004 (#4100)</td></tr>
<tr><td>CR4405:</td><td>C (#12)</td></tr>
<tr><td>CR5000:</td><td>D (#13)</td></tr>
<tr><td>CR5000RTC:</td><td>D (#13)</td></tr>
<tr><td>CR6000:</td><td>1770 (#6000)</td></tr>
<tr><td>CR8000:</td><td>6</td></tr>
</table></td></tr>
</table>

Reg	Setting Name	Default (Hex)	Comment
21C	In Stand Extended Duplicate Block Time (ms)	5DC (#1500) CR5000: CR5000RTC: 0	**Valid Range:** 0 to 7FFFFFF Milliseconds Applies to a Reader in a stand. Consecutive duplicate barcodes (i.e. barcodes that contain the same data) are blocked for this amount of time (in milliseconds). 0 turns off blocking of duplicate barcodes. **Note:** Not used in the CR5000 & CR5000RTC reader **Keywords:** #DuplicateBlock
21D	USB Speed	0	0: High Speed 1: Full Speed Speed at which USB Connection is made. **Keywords:** #Communications
235	In-Charger Event	0 CR2300: CR2600: CR2600XHD: CR3600: CR3600DPM: CR4405: E0 (#224)	This event will be triggered when the reader is placed in its charger base. See register C4 for a list of events **Keywords:** #Charger
236	Out-of-Charger Event	0 CR2300: CR2600: CR2600XHD: CR3600: CR3600DPM: CR4405: DF (#223)	This event will be triggered when the reader is removed from its charger base. See register C4 for a list of events **Keyword:** #Charger

Reg	Setting Name	Default (Hex)	Comment
243	USB Keyboard Connection Delay (ms)	0 CR2300: CR2600: CR2600XHD: CR3600: CR3600DPM: CR4405: 1388 (#5000)	**Valid Range:** 0 to 7FFFFFFF (#2147483647) Milliseconds This is the additional amount of time that the reader will wait after connecting before sending data. The delay will only be used with the associated communication mode. **Keywords:** #Communications
244	USB HID Native Connection Delay (ms)	0	**Valid Range:** 0 to 7FFFFFFF (#2147483647) Milliseconds This is the additional amount of time that the reader will wait after connecting before sending data. The delay will only be used with the associated communication mode. **Keywords:** #Communications
245	USB VCOM Connection Delay (ms)	0	**Valid Range:** 0 to 7FFFFFFF (#2147483647) Milliseconds This is the additional amount of time that the reader will wait after connecting before sending data. The delay will only be used with the associated communication mode. **Keywords:** #Communications
246	USB HIDPOS Connection Delay (ms)	0	**Valid Range:** 0 to 7FFFFFFF (#2147483647) Milliseconds This is the additional amount of time that the reader will wait after connecting before sending data. The delay will only be used with the associated communication mode. **Keywords:** #Communications
247	Bluetooth® Connection Delay (ms)	0	**Valid Range:** 0 to 7FFFFFFF (#2147483647) Milliseconds This is the additional amount of time that the reader will wait after connecting before sending data. The delay will only be used with the associated communication mode. **Keywords:** #Communications
249	Serial UART Connection Delay (ms)	0	**Valid Range:** 0 to 7FFFFFFF (#2147483647) Milliseconds This is the additional amount of time that the reader will wait after connecting before sending data. The delay will only be used with the associated communication mode. **Keywords:** #Communications

<table>
<tr><th>Reg</th><th>Setting Name</th><th>Default (Hex)</th><th>Comment</th></tr>
<tr><td>24A</td><td>Han Xin Barcode</td><td>0</td><td>Binary Dip Switch
<table>
<tr><th>Bit</th><th>Controls</th><th>Value</th></tr>
<tr><td rowspan="2">0</td><td rowspan="2">Enable/Disable Han Xin</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
</table>
Note: This register has no effect on a CR900FD without the proper 2D barcode decoding license.
Keywords: #HanXin</td></tr>
<tr><td>24F</td><td>Decoding Extras</td><td>0</td><td>This is a bitmask that enables additional decode data post-processing.
Binary Dip Switch</td></tr>
</table>

Bit	Controls	Value
0	Send Code 39 start and stop delimiters **Keywords:** #Code39	0: Disabled 1: Enabled
1	Remove Codabar start and stop delimiters **Keywords:** #Codabar	0: Disabled 1: Enabled
2	Forces all decode data (excluding AIM identifier and other prefix/suffix info) to upper case **Keywords:** #AIMId	0: Disabled 1: Enabled
3	Forces all decode data (excluding AIM identifier and other prefix/suffix info) to lower case **Keywords:** #AIMId	0: Disabled 1: Enabled
4	Removes UPC-A check digit **Keywords:** #UPC	0: Disabled 1: Enabled
5	Removes UPC-A number system digit **Keywords:** #UPC	0: Disabled 1: Enabled
6	Removes UPC-E check digit **Keywords:** #UPC	0: Disabled 1: Enabled
7	Removes UPC-E number system digit **Keywords:** #UPC	0: Disabled 1: Enabled
8	Removes EAN/JAN-13 check digit **Keywords:** #EAN/JAN	0: Disabled 1: Enabled
9	Removes EAN/JAN-8 check digit **Keywords:** #EAN/JAN	0: Disabled 1: Enabled
10	Convert EAN-8 to EAN-13 **Keywords:** #UPC	0: Disabled 1: Enabled
11	Convert UPC-A to EAN-13 **Keywords:** #UPC	0: Disabled 1: Enabled
12	Convert Bookland EAN-13 to ISBN **Keywords:** #EAN/JAN	0: Disabled 1: Enabled
13	Convert Bookland EAN-13 to ISSN **Keywords:** #EAN/JAN	0: Disabled 1: Enabled
14	Send Tri-Optic Code 39 start and stop delimiters **Keywords:** #Code39	0: Disabled 1: Enabled
15	Remove GS1 DataBar “()” characters, if present **Keywords:** #GS1Databar	0: Disabled 1: Enabled
16	Output Telepen as ASCII. (If Disabled, the output is Numeric) **Keywords:** #Telepen	0: Disabled 1: Enabled
17		0: Disabled

<table>
<tr><th>Reg</th><th>Setting Name</th><th>Default (Hex)</th><th>Comment</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td></td><td>Output Royal Mail check character
<b>Keywords:</b> #Postal</td><td>1: Enabled</td></tr>
<tr><td rowspan="2">18</td><td rowspan="2">Remove error correction code words from Australia Post output
<b>Keywords:</b> #Postal</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
</table>
</td></tr>
<tr><td>250</td><td>Korean Post Symbology</td><td>0</td><td>Binary Dip Switch
<table>
<tr><th>Bit</th><th>Controls</th><th>Value</th></tr>
<tr><td rowspan="2">0</td><td rowspan="2">Enable Korean Post</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
</table>
<b>Keywords:</b> #Postal</td></tr>
<tr><td>251</td><td>Background Transmit Enable</td><td>1
CR4405:
0</td><td>0: Disabled
1: Enabled
If Enabled, use background transmit feature.
With this enabled, user may be able to scan barcode labels faster than the reader or modem can output the data to the host.</td></tr>
</table>

<table>
<tr><th>Reg</th><th>Setting Name</th><th>Default (Hex)</th><th>Comment</th></tr>
<tr><td>256</td><td>Decoding Options</td><td>10
(#16)</td><td>CortexDecoder decoding options.
Binary Dip Switch
<table>
<tr><th>Bit</th><th>Controls</th><th>Value</th></tr>
<tr><td>1</td><td>Basic Laser DPM
<b>Keywords:</b> #DPM</td><td>0: Disabled
1: Enabled</td></tr>
<tr><td>2</td><td>Basic inkjet/dot peen DPM
<b>Keywords:</b> #DPM</td><td>0: Disabled
1: Enabled</td></tr>
<tr><td>4</td><td>Low Contrast 1D</td><td>0: Disabled
1: Enabled</td></tr>
<tr><td>5</td><td>Dot peen dark on light (need DPM license) <b>Keywords:</b> #DPM</td><td>0: Disabled
1: Enabled</td></tr>
<tr><td>6</td><td>Dot peen light on dark (DPM license required) <b>Keywords:</b> #DPM</td><td>0: Disabled
1: Enabled</td></tr>
<tr><td>7</td><td>Laser/chemical etch
(DPM license required)
<b>Keywords:</b> #DPM</td><td>0: Disabled
1: Enabled</td></tr>
<tr><td>11</td><td>Robust Inkjet DPM
<b>Keywords:</b> #DPM</td><td>0: Disabled
1: Enabled</td></tr>
<tr><td>12</td><td>Light DPM (fast reading of all 1D/2D and high contrast DPM)
<b>Keywords:</b> #DPM</td><td>0: Disabled
1: Enabled</td></tr>
</table>
<b>Note:</b> All other bits are Reserved</td></tr>
<tr><td>25B</td><td>Bluetooth® Keyboard Discoverable Time (sec)</td><td>1E
(#30)</td><td><b>Valid Range:</b> 0 to 7FFFFFFF (#2147483647) Seconds
The time that the reader will be discoverable to other Bluetooth® devices when discoverable command is sent to the reader (see Section 10.1for more info on discoverable command)</td></tr>
<tr><td>25D</td><td>Disconnect when Set Discoverable</td><td>1</td><td>0: Disabled
1: Enabled
If Enabled, disconnect when the reader is set to discoverable mode (see Section 10.1 for more info on the discoverable command)</td></tr>
</table>

<table>
<tr><th>Reg</th><th>Setting Name</th><th>Default (Hex)</th><th>Comment</th></tr>
<tr><td>25E</td><td>Connection Delay (ms) –Bluetooth® Keyboard</td><td>0</td><td><b>Valid Range:</b> 0 to 7FFFFFFF (#2147483647) Milliseconds
This is the additional amount of time that the reader will wait after connecting before sending data. The delay will only be used with the associated communication mode.</td></tr>
<tr><td>262</td><td>Define decode data encoding and conversion method</td><td>0</td><td>Used to specify the input encoding of the data being decoded. The data will either be used as is or be converted to Unicode code points for the specified input encoding, based on the setting below.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Define that the input encoding is ASCII – do not convert</td></tr>
<tr><td>1</td><td>Define that the input encoding is ASCII and convert the ASCII to Unicode code points</td></tr>
<tr><td>2</td><td>Define that the input encoding is UTF-8 and convert the UTF-8 to Unicode code points</td></tr>
</table>
<b>Keywords:</b> #DataEncoding; #KeyboardMap</td></tr>
<tr><td>270</td><td>Requested Download Memory Allocation Size</td><td>0</td><td>This is the optimal amount of memory to allocate for the current file download to the reader.
<b>Persistence:</b> Reset-on-boot</td></tr>
<tr><td>271</td><td>Connection Delay (ms) –iAP2 External Accessory mode</td><td>0</td><td><b>Valid Range:</b> 0 to 7FFFFFFF (#2147483647) Milliseconds
This is the additional amount of time that the reader will wait after connecting before sending data. The delay will only be used with the associated communication mode.</td></tr>
<tr><td>272</td><td>Connection Delay (ms) –iAP2 Keyboard mode</td><td>0</td><td><b>Valid Range:</b> 0 to 7FFFFFFF (#2147483647) Milliseconds
This is the additional amount of time that the reader will wait after connecting before sending data. The delay will only be used with the associated communication mode.</td></tr>
<tr><td>273</td><td>Max Error Log Size (bytes)</td><td>C8 * 1000 ((#200) * (#1024))</td><td>This is the maximum size of the error log in bytes. It will not exceed it by more than the size of one error log entry.</td></tr>
<tr><td>274</td><td>Error Log Truncation Percent</td><td>46
(#70)</td><td><b>Valid Range:</b> 0 to 64 (#100) Percent
When the error log exceeds the maximum length, it will be truncated to this percentage of its maximum size</td></tr>
</table>

Reg	Setting Name	Default (Hex)	Comment
277	Max Motion Brightness Percent	3C (#60) CR5000: CR5000RTC: 64 (#100)	**Valid Range:** 0 to 64 (#100) Percent The percentage of maximum illumination of the red LED when used in motion detection. **Keywords:** #MotionDetection
278	Low Battery Sleep Voltage (mV)	0 CR4405: C1C (#3100)	The reader will automatically sleep when the battery voltage is below this threshold. **Note:** Supported on CR4405 only. **Keywords:** #Battery
279	Low Battery Sleep Wakeup Voltage (mV)	0 CR4405: ED8 (#3800)	If the reader automatically sleeps because of a low battery, it will wake automatically if the battery voltage goes above this level. **Note:** Supported on CR4405 only. **Keywords:** #Battery
27B	Cabled Event	0 CR4405: E1 (#225)	This event will be triggered when the reader is connected via a cable. See register C4 for a list of events **Note:** Supported on CR4405 only.
27C	Uncabled Event	0 CR4405: DE (#222)	This event will be triggered when the reader is disconnected from its cable. See register C4 for a list of events **Note:** Supported on CR4405 only.
27D	Phone Connected Event	DC (#220)	This event will be triggered when a phone is connected to the reader. See register C4 for a list of events **Note:** Supported on CR4405 only.
27E	Phone Disconnected Event	DD (#221)	This event will be triggered when a phone is disconnected from the reader. See register C4 for a list of events **Note:** Supported on CR4405 only.

Reg	Setting Name	Default (Hex)	Comment
27F	Low Battery Level (%)	A (#10)	**Valid Range:** 0 to 64 (#100) Percent Used for battery status indication **Note:** Supported on CR4405 only. **Keywords:** #Battery
280	Medium Battery Level (%)	1E (#30)	**Valid Range:** 0 to 64 (#100) Percent Used for battery status indication **Note:** Supported on CR4405 only. **Keywords:** #Battery
282	Sled Battery Critical Level (%)	F (#15)	**Valid Range:** 0 to 64 (#100) Percent Reader will not supply current to a connected phone below this level. This applies to all power management schemes. **Note:** Supported on CR4405 only. **Keywords:** #Battery
284	Device Battery Charge Mode	2	Defines the way in which the reader will attempt to charge a connected device/phone. All power schemes involving charging the device will stop charging when the critical sled battery level is reached (register 282). 0: Do not charge 1: Keep phone fully charged 2: Managed Power Scheme: • Never charge device (phone) if sled battery is below value in register 282 (Sled battery critical level). This always trumps other conditions that would normally initiate device (phone) charging. • Assuming first condition is not true, always charge phone until sled battery drops below value in register 29D. This is the reserve amount of sled battery required to maintain desired scanner usage (Typically ~12 hours, 100 scans/hr). • Charge phone if device battery drops below value in register 29C. Cancel charging when device battery rises above the value in register 29C added to the value in register 285. **Note:** Supported on CR4405 only. **Keywords:** #Battery

Reg	Setting Name	Default (Hex)	Comment
285	Device Battery Delta To Stop Charge (%)	5	**Valid Range:** 0 to 64 (#100) Percent Applies when the Device Battery Charge Mode (284) is set to Managed Power Scheme (the value 2). If sled is charging the device due to the Device Battery Critical Level (29C) being met, the device battery level will need to rise above the Device Battery Maintenance Level (29B) by this percent before sled disables charging. **Note:** Supported on CR4405 only. **Keywords:** #Battery
286	Device Battery Update Max Interval (ms)	493E0 (#300000)	**Valid Range:** 0 to 7FFFFFFF (#2147483647) Milliseconds If this amount of time (ms) elapses without a device battery update, only the sled battery percentage is used to determine whether the device charging state should be changed. **Note:** Supported on CR4405 only. **Note:** Applies to Device Battery Charge Mode (284) option 2 only **Keywords:** #Battery
288	Full Battery Level (%)	5F (#95)	**Valid Range:** 0 to 64 (#100) Percent Used for battery status indication **Note:** Supported on CR4405 only. **Keywords:** #Battery
289	Low Battery Pre Indicate Period (ms)	1F4 (#500)	**Valid Range:** 0 to 7FFFFFFF (#2147483647) Milliseconds When indicating a low battery (and not charging), this is the time between the triggering event (such as a button press) and the first blink. **Note:** Supported on CR4405 only. **Keywords:** #Battery
28A	Low Battery Indicate Period (ms)	1F4 (#500)	**Valid Range:** 0 to 7FFFFFFF (#2147483647) Milliseconds When indicating a low battery (and not charging), this is the blink duration. **Note:** Supported on CR4405 only. **Keywords:** #Battery

Reg	Setting Name	Default (Hex)	Comment
28B	Low Battery Post Indicate Period (ms)	1F4 (#500)	**Valid Range:** 0 to 7FFFFFFF (#2147483647) Milliseconds When indicating a low battery (and not charging), this is the time between blinks. **Note:** Supported on CR4405 only. **Keywords:** #Battery
28C	Low Battery Indicate Pulses	3	When indicating a low battery (and not charging), this is the number of blinks that will occur due to a triggering event (such as a button press). **Note:** Supported on CR4405 only. **Keywords:** #Battery
28D	Medium Battery Pre Indicate Period (ms)	1F4 (#500)	**Valid Range:** 0 to 7FFFFFFF (#2147483647) Milliseconds When indicating a medium battery (and not charging), this is the time between the triggering event (such as a button press) and the first blink. **Note:** Supported on CR4405 only. **Keywords:** #Battery
28E	Medium Battery Indicate Period (ms)	1F4 (#500)	**Valid Range:** 0 to 7FFFFFFF (#2147483647) Milliseconds When indicating a medium battery (and not charging), this is the blink duration. **Note:** Supported on CR4405 only. **Keywords:** #Battery
28F	Medium Battery Post Indicate Period (ms)	1F4 (#500)	**Valid Range:** 0 to 7FFFFFFF (#2147483647) Milliseconds When indicating a medium battery (and not charging), this is the time between blinks. **Note:** Supported on CR4405 only. **Keywords:** #Battery
290	Medium Battery Indicate Pulses	3	When indicating a medium battery (and not charging), this is the number of blinks that will occur due to a triggering event (such as a button press). **Note:** Supported on CR4405 only. **Keywords:** #Battery

Reg	Setting Name	Default (Hex)	Comment
291	Low Battery Charging Indicate Period (ms)	1F4 (#500)	**Valid Range:** 0 to 7FFFFFFF (#2147483647) Milliseconds When indicating a low battery while charging, this is the duration of each LED color display in the color sequence (see register 297). **Note:** Supported on CR4405 only. **Keywords:** #Battery
292	Medium Battery Charging Indicate Period (ms)	1F4 (#500)	**Valid Range:** 0 to 7FFFFFFF (#2147483647) Milliseconds When indicating a medium battery while charging, this is the duration of each LED color display in the color sequence (see register 298). **Note:** Supported on CR4405 only. **Keywords:** #Battery
293	High Battery Charging Indicate Period (ms)	1F4 (#500)	**Valid Range:** 0 to 7FFFFFFF (#2147483647) Milliseconds When indicating a high (but not yet fully charged) battery while charging, this is the duration of each LED color display in the color sequence (see register 299). **Note:** Supported on CR4405 only. **Keywords:** #Battery
294	Low Battery Charging Post Indicate Period (ms)	1F4 (#500)	**Valid Range:** 0 to 7FFFFFFF (#2147483647) Milliseconds When indicating a low battery while charging, this is the time between LED color sequences (which are defined in register 297). **Note:** Supported on CR4405 only. **Keywords:** #Battery
295	Medium Battery Charging Post Indicate Period (ms)	1F4 (#500)	**Valid Range:** 0 to 7FFFFFFF (#2147483647) Milliseconds When indicating a medium battery while charging, this is the time between LED color sequences (which are defined in register 298). **Note:** Supported on CR4405 only. **Keywords:** #Battery

Reg	Setting Name	Default (Hex)	Comment
296	High Battery Charging Post Indicate Period (ms)	1F4 (#500)	**Valid Range:** 0 to 7FFFFFFF (#2147483647) Milliseconds When indicating a high (but not yet fully charged) battery while charging, this is the time between LED color sequences (which are defined in register 299). **Note:** Supported on CR4405 only. **Keywords:** #Battery, #Charging

297	Low Battery Charging Indicate Color	1	Color to display when charging the battery and the battery is low. 1: red 2: yellow 4: green All other values are invalid. **Note:** Supported on CR4405 only. **Keywords:** #Battery, #Charging

Reg	Setting Name	Default (Hex)	Comment
298	Medium Battery Charging Indicate Color	2	Color to display when charging the battery and the battery is medium. 1: red 2: yellow 4: green All other values are invalid. **Note:** Supported on CR4405 only. **Keywords:** #Battery, #Charging

Reg	Setting Name	Default (Hex)	Comment
299	High Battery Charging Indicate Color	4	Color to display when charging the battery and the battery is high. 1: red 2: yellow 4: green All other values are invalid. **Note:** Supported on CR4405 only. **Keywords:** #Battery, #Charging
29B	Device Battery Maintenance Level (%)	64 (#100)	**Valid Range:** 0 to 64 (#100) Percent Only applies to charging mode 3 in charging mode register 284. Value at which charging scheme attempts to keep the device battery level. **Note:** Supported on CR4405 only. **Keywords:** #Battery
29C	Device Battery Critical Level (%)	19 (#25)	**Valid Range:** 0 to 64 (#100) Percent Only applies to charging mode 2 in charging mode register 284. If device battery drops below this level, sled will start charging the device. **Note:** Supported on CR4405 only. **Keywords:** #Battery

Reg	Setting Name	Default (Hex)	Comment
29D	Sled Battery Reserve Level (%)	28 (#40)	**Valid Range:** 0 to 64 (#100) Percent Only applies to charging mode 2 in charging mode register 284. Sled will charge device battery until sled battery drops below the percentage value in this register. **Note:** Supported on CR4405 only. **Keywords:** #Battery
29E	Lower battery percent scaling (%)	0	**Valid Range:** 0 to 64 (#100) Percent Reader will report 0% battery capacity if the actual battery drops below this percent. **Keywords:** #Battery
29F	Upper battery percent scaling (%)	64 (#100) CR4405: 5A (#90)	**Valid Range:** 0 to 64 (#100) Percent Reader will report 100% battery capacity if the actual battery rises above this percent. **Keywords:** #Battery
2A3	Code 49 Symbology	0	0: Disabled 1: Enabled
2A8	Use demo script	0 CR3600: 1	0: Disabled 1: Enabled If Enabled on a device that has a display, this setting will cause the device to use the demo script instead of the ".default.js" script. **Persistence:** Protected
2AB	Detect charging base as Stand	0	0: Disabled 1: Enabled If Enabled, for readers with a charging base, the reader will be considered In-Stand if the reader is in the charging base. **Note:** Applies to Bluetooth® readers only. **Keywords:** #Stand, #StandDetect

Reg	Setting Name	Default (Hex)	Comment
2AF	USB pass-through mode minimum battery capacity percent	5	**Valid Range:** 0 to 64 (#100) Percent When in USB pass-through mode, if the sled battery drops to or below this level, the sled will stop providing power to the iPhone. **Note:** This will cause a communications failure between the iPhone and the USB host, but it is necessary to prevent sled reboots that would also cause communications failures. **Note:** Supported on CR4405 only. **Keywords:** #Battery
2B0	USB pass-through mode hysteresis battery capacity percent	19 (#25)	**Valid Range:** 0 to 64 (#100) Percent When in USB pass-through mode, if the sled battery drops below the USB pass-through mode minimum battery capacity, the sled battery must reach this higher capacity before power (and communication) will be restored to the iPhone. **Note:** Supported on CR4405 only. **Keywords:** #Battery

<table>
<tr><th>Reg</th><th>Setting Name</th><th>Default (Hex)</th><th>Comment</th></tr>
<tr><td>2B2</td><td>Illumination Axis Control</td><td>0
CR5000:
CR5000RTC:
70
(#112)</td><td>Internal illumination control
For CR5000 & CR5000RTC:
<table>
<tr><th>Bit</th><th>Controls</th><th>Value</th></tr>
<tr><td>0-2</td><td>Unused</td><td></td></tr>
<tr><td rowspan="4">3-4</td><td rowspan="4">CR5000 & CR5000RTC quick decode behavior</td><td>0: Invalid</td></tr>
<tr><td>1: Use red illumination</td></tr>
<tr><td>2: Use IR illumination</td></tr>
<tr><td>3: Use red and IR illumination</td></tr>
<tr><td rowspan="4">5-6</td><td rowspan="4">CR5000 & CR5000RTC standard decode behavior</td><td>0: Use the same as the CR5000 & CR5000RTC quick decode behavior</td></tr>
<tr><td>1: Use red illumination + fixed 20% IR</td></tr>
<tr><td>2: Use red illumination + fixed 30% IR</td></tr>
<tr><td>3: Use red illumination only</td></tr>
</table>
<b>Note:</b> Applies to CR5000 & CR5000RTC only. All other readers will ignore this register.
<b>Keywords:</b> #AGC, #AGCMode, #DPM</td></tr>
<tr><td>2B4</td><td>Public Sector Behavior</td><td>0</td><td>Binary Dip Switch
<table>
<tr><th>Bit</th><th>Controls</th><th>Value</th></tr>
<tr><td rowspan="2">0</td><td rowspan="2">Enable public-sector parsing</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">1</td><td rowspan="2">Enable JSON formatting</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">2</td><td rowspan="2">Reject raw data (non-public-sector-formatted decodes are considered an error condition)</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">3</td><td rowspan="2">Enable Stand Alone Age Verification (CR5000RTC only)</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
</table>
<b>Note:</b> Bit 0, 1, & 2 are for DL Formatting and require license 5014. Bit 3 is for Stand Alone Age Verification (CR5000RTC only) and require license 5017.
<b>Keywords:</b> #AgeVerification</td></tr>
</table>

Reg	Setting Name	Default (Hex)	Comment
2B6	Trigger 1 Release Event	0	If non-zero, the specified event is generated when this button is released. **Note:** Register C4 contains the list of valid events. **Keywords:** #DefaultEvent
2B7	Trigger 2 Release Event	0	If non-zero, the specified event is generated when this button is released. **Note:** Register C4 contains the list of valid events. **Keywords:** #DefaultEvent
2B8	Trigger 3 Release Event	0	If non-zero, the specified event is generated when this button is released. **Note:** Register C4 contains the list of valid events. **Keywords:** #DefaultEvent
2B9	Trigger 4 Release Event	0	If non-zero, the specified event is generated when this button is released. **Note:** Register C4 contains the list of valid events. **Keywords:** #DefaultEvent
2BA	Trigger 5 Release Event	0	If non-zero, the specified event is generated when this button is released. **Note:** Register C4 contains the list of valid events. . **Keywords:** #DefaultEvent
2BB	Trigger 6 Release Event	0	If non-zero, the specified event is generated when this button is released. **Note:** Register C4 contains the list of valid events. **Keywords:** #DefaultEvent
2BC	Trigger 7 Release Event	0	If non-zero, the specified event is generated when this button is released. **Note:** Register C4 contains the list of valid events. **Keywords:** #DefaultEvent
2BD	Allow Repeated JavaScript Button Events	0	0: If the button event is a JavaScript event, the button event will only be generated once. 1: If the button event is a JavaScript event, the button event will continue to be generated until the button is released.

Reg	Setting Name	Default (Hex)	Comment
2BE	In-Stand Event	0 CR2300: CR2600: CR2600XHD: CR3600: CR3600DPM: F2 (#242) CR5000: CR5000RTC: F4 (#244)	This event will be posted when the reader is placed in its stand (for CR5000 & CR5000RTC, when it is placed on a surface). See register C4 for a list of events. **Note:** Applies to Bluetooth® and CR5000 & CR5000RTC readers only. **Keywords:** #Stand, #StandModes, #StandDetect, #DefaultEvent
2BF	Out-of-Stand Event	0 CR2300: CR2600: CR2600XHD: CR3600: CR3600DPM: F2 (#242) CR5000: CR5000RTC: F5 (#245)	This event will be posted when the reader is removed from its stand. See register C4 for a list of events. **Note:** Applies to Bluetooth® and CR5000 & CR5000RTC readers only. **Keywords:** #Stand, #StandModes, #StandDetect, #DefaultEvent
2C0	In-Stand Target Tolerance	640 (#1600)	When the reader is placed on a surface and Stand Detection is enabled, register 9D will be set to the value in this register. **Note:** Applies to CR5000 & CR5000RTC readers only. **Keywords:** #Stand, #StandModes
2C1	Out-of-Stand Target Tolerance	640 (#1600)	When the reader is lifted up and Stand Detection is enabled, register 9D will be set to the value in this register. **Note:** Applies to CR5000 & CR5000RTC readers only. **Keywords:** #Stand, #StandModes

Reg	Setting Name	Default (Hex)	Comment
2C2	In-Stand Default Event	FF (#255) CR2300: CR2600: CR2600XHD: CR3600: CR3600DPM: F0 (#240) CR5000: CR5000RTC: 6	When the reader is placed in the stand (for CR5000 & CR5000RTC, when it is placed on a surface) and Stand Detection is enabled, register C4 will be set to the value in this register. See register C4 for a list of events. **Note:** Applies to Bluetooth® and CR5000 & CR5000RTC readers only. Also see registers: 1F7 to enable Stand Detection, 2AB to define the charging base as a 'Stand' and C4 for a list of events **Keywords:** #DefaultEvent, #StandModes
2C3	Out-of-Stand Default Event	FF (#255) CR5000: CR5000RTC: 2	When the reader is removed from the stand (for CR5000 & CR5000RTC, when it is lifted up) and Stand Detection is enabled, register C4 will be set to the value in this register. See register C4 for a list of events. **Note:** Applies to Bluetooth® and CR5000 & CR5000RTC readers only. Also see registers: 1F7 to enable Stand Detection, 2AB to define the charging base as a 'Stand' and C4 for a list of events **Keywords:** #DefaultEvent, #StandModes
2C4	Minimum Motion Illumination Percent	1 CR3600DPM: CR6000: 3	**Valid Range:** 0 to 64 (#100) Percent This is the minimum illumination that will be used when in motion detection mode. Also see register: 206 **Keywords:** #MotionDetection
2C5	Minimum Initial Battery Percent to power iPhone	5	**Valid Range:** 0 to 64 (#100) Percent In iPhone communication modes (Apple® App mode and Apple® Keyboard mode, currently) the reader will not provide any power or communication channel to the iPhone until it has confirmed that the sled battery is above this percent. **Note:** Supported on CR4405 only. **Keywords:** #Battery

Reg	Setting Name	Default (Hex)	Comment
2C6	Keyboard minimum inter-character delay (ms)	0	**Valid Range:** 0; 3 to 7FFFFFFF (#2147483647) Milliseconds Require at least this amount of time between characters sent over a keyboard interface. Any value below 3 ms will have no noticeable effect. **Keywords:** #InterCharacterDelay
2C7	Keyboard minimum inter-scancode delay (non-release scancode) (ms)	0	**Valid Range:** 0; 3 to 7FFFFFFF (#2147483647) Milliseconds Require at least this amount of time since the last scancode sent over a keyboard interface (this setting is used when the scancode to be sent is not a release scancode). Any value below 3 ms will have no noticeable effect. **Keywords:** #InterCharacterDelay
2C8	Keyboard minimum inter-scancode delay (release scancode) (ms)	0	**Valid Range:** 0; 3 to 7FFFFFFF (#2147483647) Milliseconds Require at least this amount of time since the last scancode sent over a keyboard interface (this setting is used when the scancode to be sent is a release scancode). Any value below 3 ms will have no noticeable effect. **Keywords:** #InterCharacterDelay
2C9	Enable vibrate in stand	0	0: Disabled 1: Enabled If Enabled, this setting will allow the reader to vibrate when in the stand. The default is to prevent vibration of the reader while in the stand. **Note:** Readers with a vibration motor are the CR1400, CR1400XHD, CR1428, CR2600, CR2600XHD, CR3600, CR3600DPM and CR6000. **Keywords:** #Stand, #Vibration

<table>
<tr><th>Reg</th><th>Setting Name</th><th>Default (Hex)</th><th>Comment</th></tr>
<tr><td>2CA</td><td>Extended Duplicate Block Time Options</td><td>1</td><td>Binary Dip Switch
<table>
<tr><th>Bit</th><th>Controls</th><th>Value</th></tr>
<tr><td rowspan="2">0</td><td rowspan="2">Use extended duplicate block time if stand is detected</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">1</td><td rowspan="2">Use extended duplicate block time if in motion detection mode</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">2</td><td rowspan="2">Use extended duplicate block time if in continuous scan mode</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">3</td><td rowspan="2">Always use extended duplicate block time</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
</table>
<b>Keywords:</b> #DuplicateBlock, #Stand</td></tr>
<tr><td>2CB</td><td>iPhone Onscreen Keyboard Release Delay (ms)</td><td>7D0
(#2000)</td><td><b>Valid Range:</b> 0 to 7FFFFFFF (#2147483647) Milliseconds
When the iPhone receives keyboard data from the sled, the onscreen keyboard will disappear before sending the first character, and reappear after receiving the last character. This setting is the number of milliseconds that the onscreen keyboard will remain hidden after receiving the last character from the sled.
<b>Note:</b> Supported on CR4405 only.</td></tr>
<tr><td>2CC</td><td>Allow sending USB scancodes over a Bluetooth® SPP connection</td><td>1</td><td>0: Disabled
1: Enabled
If Enabled, the reader will send data as USB scancodes to a modem that requests this behavior, instead of sending just the decoded bytes. If Disabled, or if the modem does not request USB scancodes, the reader will send the data as the decoded bytes.
<b>Note:</b> Supported on Bluetooth® readers only.</td></tr>
</table>

<table>
<tr>
<td>2CD</td>
<td>“On QuickConnect Barcode” behavior</td>
<td>0</td>
<td>This setting defines special behavior that will occur when a QuickConnect Barcode (QCB) is scanned. A QCB is defined as any barcode that contains the “:%07” command.
<table>
<tr><th>Bits</th><th>Controls</th></tr>
<tr><td>0-15</td><td>QuickConnect Barcode special operation</td></tr>
<tr><td>16-30</td><td>QuickConnect Barcode parameter</td></tr>
</table>
<b>QuickConnect Barcode special operation:</b>
0 = No special operation/behavior
1 = Override communication protocol in 1-way mode
2 = Override communication protocol in 2-way mode
3 = Post an event

<b>QuickConnect Barcode parameter:</b>
For special operations 1 and 2, this parameter is the desired communication protocol (see register 08).
For special operation 3, this parameter is the event to post (see register C4).

For example, to force the reader into “raw” communication protocol mode when scanning a QCB, set this register to 0x10001. To post the 0x40 JavaScript event after scanning a QCB, set this register to 0x400003.</td>
</tr>
</table>

<table>
<tr><th>Reg</th><th>Setting Name</th><th>Default (Hex)</th><th>Comment</th></tr>
<tr><td>2CE</td><td>Motion Detection Pattern Option</td><td>1
CR5000:
CR5000RTC:
5</td><td>This setting defines the type of pattern detection to be applied after initial motion is detected.
<table>
<tr><th>Bit</th><th>Controls</th><th>Value</th></tr>
<tr><td rowspan="2">0</td><td rowspan="2">Basic pattern detection. It has fast response time and can be used for general applications.</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">1</td><td rowspan="2">Advance pattern detection. It has better detection for low contrast labels and 2D barcodes but is slower.</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">2</td><td rowspan="2">Cellphone detection (CR5000 & CR5000RTC only)</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
</table>
<b>Note:</b>
<ul>
<li>Without applying any pattern detection (value as 0), the reader will capture an image for decoding when any motion is detected.</li>
<li>When Basic pattern detection is not sensitive enough for some cases, you can turn both Basic and Advance pattern detection ON (value as 3). In this situation, most of the cases will be detected by the Basic pattern detection with fast response time and the Advance pattern detection can detect low contrast labels and some 2D barcodes.</li>
</ul>
<b>Keywords:</b> #Cellphone, #MotionDetection</td></tr>
<tr><td>2CF</td><td>Re-sleep Timer (ms)</td><td>FFFFFFFFFFFFFE0C
(#-500)
(disabled)
CR4405: 45000
(#282624)</td><td><b>Valid Range:</b> 0 to 7FFFFFFF (#2147483647) Milliseconds; FFFFFFFFFFFFFE0C (#-500) (Disabled)
If the auto wake feature is enabled (register 2D0: Full Wake Duration > 0), when the reader auto-wakes from low power sleep mode, it will stay awake for this amount of time (ms) before going back to low power mode. This assumes the user does not perform some other action that would “wake” the reader normally, like press a button.
<b>Keywords:</b> #ReaderState</td></tr>
</table>

Reg	Setting Name	Default (Hex)	Comment
2D0	Full Wake Period (sec)	FFFFFFFFFFFFFE0C (#-500) (disabled) CR4405: 384 (#900) (15 min)	**Valid Range:** 0 to 7FFFFFFF (#2147483647) Seconds; FFFFFFFFFFFFFE0C (#-500) (Disabled) If greater than zero, this is the amount of time (sec) that the reader will remain asleep before auto-waking for the duration defined in register 2CF. If zero or negative, feature will be disabled and the reader will never auto-wake. **Keywords:** #ReaderState
2D1	Storage Full Threshold (percent)	5A (#90)	**Valid Range:** 0 to 5A (#90) Percent This setting limits the available storage to a percentage of the total storage. It is not recommended to increase this threshold greater than 90%.
2D3	Control Character Input Method	0	This setting defines the input method in which Control Characters are sent when in keyboard mode. 0: Language Default 1: Ctrl + Character 2: Alt + Keypad 3: Alt + Leading Zero **Persistence:** Preserved **Note:** The language default is defined by the language XML file **Keywords:** #KeyboardMap
2DB	In-Stand Target Enable	0	0: Disabled 1: Enabled When the reader is placed on a surface and Stand Detection is enabled, register F will be set to the value in this register. **Note:** Applies to CR5000 & CR5000RTC readers only. **Keywords:** #Stand, #StandModes
2DC	Out-of-Stand Target Enable	0	0: Disabled 1: Enabled When the reader is lifted up and Stand Detection is enabled, register F will be set to the value in this register. **Note:** Applies to CR5000 & CR5000RTC readers only. **Keywords:** #Stand, #StandModes

Reg	Setting Name	Default (Hex)	Comment
2D D	In-Stand Illumination Axis Control	0 CR5000: CR5000RTC: 70 (#112)	When the reader is placed on a surface and Stand Detection is enabled, register 2B2 will be set to the value in this register. **Note:** Applies to CR5000 & CR5000RTC readers only. **Keywords:** #Stand, #StandModes
2DE	Out-of-Stand Illumination Axis Control	0 CR5000: CR5000RTC: 70 (#112)	When the reader is lifted up and Stand Detection is enabled, register 2B2 will be set to the value in this register. **Note:** Applies to CR5000 & CR5000RTC readers only. **Keywords:** #Stand, #StandModes
2DF	In-Stand Ignore Duplicate Barcode (ms)	258 (#600)	**Valid Range:** 0 to 7FFFFFFF (#2147483647) Milliseconds When the reader is placed on a surface and Stand Detection is enabled, register 159 will be set to the value in this register. The CR5000 & CR5000RTC does not use register 21C (the extended duplicate block time). **Note:** Applies to CR5000 & CR5000RTC readers only. **Keywords:** #Stand, #StandModes
2E0	Out-of-Stand Ignore Duplicate Barcode (ms)	0	When the reader is lifted up and Stand Detection is enabled, register 159 will be set to the value in this register. The CR5000 & CR5000RTC does not use register 21C (the extended duplicate block time). **Note:** Applies to CR5000 & CR5000RTC readers only. **Keywords:** #Stand, #StandModes
2E1	In-Stand Illuminate-on-Motion	0	0: Disabled 1: Enabled When the reader is lifted up and Stand Detection is enabled, register 2E3 will be set to the value in this register. **Note:** Applies to CR5000 & CR5000RTC readers only. **Keywords:** #Stand, #StandModes

Reg	Setting Name	Default (Hex)	Comment
2E2	Out-of-Stand Illuminate-on-Motion	0	0: Disabled 1: Enabled When the reader is placed on a surface and Stand Detection is enabled, register 2E3 will be set to the value in this register. **Note:** Applies to CR5000 & CR5000RTC readers only. **Keywords:** #Stand, #StandModes
2E3	Illuminate-on-Motion	0	0: Disabled 1: Enabled If set to 1, switches out of “quick decode” mode for a period of time (the illuminate-on-motion time) when motion is detected. Otherwise, motion does not affect the “quick decode” behavior. **Note:** Applies to CR5000 & CR5000RTC readers only. **Keywords:** #Stand, #StandModes
2E4	Illuminate-on-Motion Duration (ms)	3E8 (#1000)	**Valid Range:** 0 to 7FFFFFFF (#2147483647) Milliseconds Period of time that “quick decoding” will be disabled and illumination will occur after detecting motion if Illuminate on Motion is enabled. **Note:** Applies to CR5000 & CR5000RTC readers only. **Keywords:** #Stand, #StandModes
2E5	Stand Alone Age Verification Primary Age	15 (#21)	**Valid Range:** 0 to 100 (decimal practical limit) The value used to indicate if the age of the holder of the document is above or below this age. **Note:** Applies to CR5000RTC readers only. **Keywords:** #AgeVerification
2E6	Stand Alone Age Verification Secondary Age	0	**Valid Range:** 0 to the value of Stand Alone Age Verification Primary Age (2E5) The value used to indicate if the age of the holder of the document is below the Stand Alone Age Verification Primary Age but above or equal to this age. The value will be ignored if 0 or larger or equal to Stand Alone Age Verification Primary Age. **Note:** Applies to CR5000RTC readers only. **Keywords:** #AgeVerification

<table>
<tr><th>Reg</th><th>Setting Name</th><th>Default (Hex)</th><th>Comment</th></tr>
<tr><td>2E7</td><td>Maximum In-Stand Brightness Percent</td><td>3C
(#60)</td><td><b>Valid Range:</b> 0 to 64 (#100) Percent
The percentage of maximum illumination of the red LED under certain conditions.
Maximum In-Stand Brightness Percent applies when the reader is 'In-Stand' but only for images with settings optimized for barcodes printed on paper and not images with settings optimized for barcodes displayed on cellphone screens. For more information on when the CR5000 & CR5000RTC uses those settings, see Cellphone Settings (231).
<b>Note:</b> Applies to CR5000 & CR5000RTC readers only.
<b>Keywords:</b> #Stand, #StandModes</td></tr>
<tr><td>2F0</td><td>Output Encoding of Unicode converted input</td><td>0</td><td>This setting determines output encoding of decoded data that was previously converted to Unicode code points. This register is ignored if no conversion was made.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Send Unicode data using Keyboard XML lookup
<b>Keywords:</b> #KeyboardMap</td></tr>
<tr><td>1</td><td>Send Unicode data using Alt-Sequences</td></tr>
</table>
<b>Note:</b> Applies to HID Keyboard modes only.
<b>Keywords:</b> #DataEncoding</td></tr>
<tr><td>2F1</td><td>Output Good Read on RTS</td><td>0</td><td>Controls whether the Serial Flow Control RTS line should be used to output a Good Read signal.
The pulse width of this signal is controlled by setting 0x01EA.
<table>
<tr><th>Bit</th><th>Controls</th><th>Value</th></tr>
<tr><td rowspan="2">0</td><td rowspan="2">Enable Output of Good Read on RTS</td><td>0: Disabled</td></tr>
<tr><td>1: Enabled</td></tr>
<tr><td rowspan="2">1</td><td rowspan="2">Good Read RTS Polarity</td><td>0: RTS Active Low on Good Read</td></tr>
<tr><td>1: RTS Active High on Good Read</td></tr>
</table>
<b>Note:</b> Undefined behavior will result if this setting is enabled when Serial Flow Control (0x1F) is enabled.
<b>Keywords:</b> #GoodReadRTS</td></tr>
</table>

Reg	Setting Name	Default (Hex)	Comment
2F2	Leave Targeting On During Read	0	0: The targeting LED will be turned off during the brief period during which the reader is capturing an image 1: The targeting LED will not be affected by the reader capturing an image
2FE	Allow Host Communication If Locked	1	0: Disabled 1: Enabled When this register is enabled, a locked reader will process commands sent directly from the host, including CortexTools, CortexMobile, or custom applications based on the CodeUtil library. This includes applications written for PC, Apple, Android and Windows Mobile. **Keywords:** #SettingsLock
2FF	Reject PDF417 in AV Mode	1	0: Disabled 1: Enabled When this register is enabled, a reader in SimpleAV mode will reject all PDF417 barcodes which the decoder determines are not valid Driver License data. **Note:** Applies to CR5000RTC readers in SimpleAV mode only. **Keywords:** #SimpleAV
300	Ignore DL Expiration Date in AV Mode	0	0: Disabled 1: Enabled When this register is enabled, a reader in SimpleAV mode will ignore the expiration date in a Driver License. This includes DL data that does not include expiration data. The current Minnesota DL is one example that does not include expiration. **Note:** Applies to CR5000RTC readers in SimpleAV mode only. **Keywords:** #SimpleAV

Reg	Setting Name	Default (Hex)	Comment
301	Allow Changing Wireless Communication Link When Locked	1	0: Disabled 1: Enabled When this register is enabled, a locked reader is allowed to change connection between wireless devices. For CR2300/CR2600/CR3600/CR3600DPM readers, this is Code Modems and other Bluetooth® devices. For CR4405, this is the Wi-Fi connection to CortexWedge®. A reader must be connected to a wireless prior to locking it if this setting is disabled – when disabled, no connection changes are allowed after the reader is locked. **Keywords:** #SettingsLock

10 Radio Commands

The Host controls the radio by issuing *':'* commands. The following tables describe the available commands. Command numbers less than 0x80 are Bluetooth® commands.

The '#' column is the radio command number (in hexadecimal) to be used with the *':'* command. For example, ":%0E" gets the Bluetooth® device address.

The '# bytes' column indicates how many bytes of data are required as arguments for the command.

10.1 Bluetooth® Commands

Name	#	Comments	# bytes
Disconnect	00	Terminate the current connection. **Example:** :%00	0
Auto Connect	07	If connection information exists for the specified Device Address, use it to establish a connection. Otherwise, attempt to establish a connection and store the resulting information. Address is sent as 12 hexadecimal characters. **Example:** :%07000BEF023555	6
Clear Setup	08	Remove connection information associated with the specified Bluetooth® Address (as 12 hexadecimal characters). **Example:** :%08000BEF023555	6

Name	#	Comments	# bytes
Send Setup	09	The modem will send all connection information in the following format: iiii xxxxxxxxxxxx p Where iiii is the storage index, xxxxxxxxxxxx is the Bluetooth® Device Address, and p indicates pairing enabled (y) or pairing disabled (n). **Example:** :%09	0
Get Bluetooth® Address	0E	Get Bluetooth® address as 12 hexadecimal characters **Example:** :%0E	0
Get "user friendly" name	0F	Get device's "user friendly" name **Example:** :%0F	0
Clear Connection History	12	Clears the connection history (Bluetooth® addresses, link keys, etc.). It is recommended that the reader be rebooted after this operation.	0
Set Bluetooth® PIN	14	Sets the Bluetooth® PIN (up to 16 characters). Default is '0000'. **Example:** : %14abcd ;set Bluetooth® PIN to 'abcd' If the pin is being used with register 173 to connect to a CodeXML Modem, the pin must be 8 characters or greater and match the value used in setting the Modem pin (see the setPin command in D018582 Modem ICD) **Keywords:** #Encryption	1-16
Get Bluetooth® PIN	15	Gets the Bluetooth® PIN **Example:** :%15	0
Set Reader Discoverable	17	Sets reader into discoverable mode for the time specified in register 25B (default 30 sec). Only applies when reader is in Bluetooth® keyboard comm mode. **Example:** :%17	0

Name	#	Comments	# bytes
Send Pin Digit	18	Used to send Bluetooth® PIN characters to the reader. Applies to Bluetooth® Keyboard comm mode. Digits must be URL encoded. For example to send an ASCII '8', the character would be '%38', so the command to send an 8 is **Send Character Example:** :%18%38 When all the characters have been sent (typically 4, 6, or 8 characters), the reader needs a "done" command to signal the pin entry process is complete. This is accomplished by sending a null character '%00' **Send Done Example:** :%18%00	1

11 Symbology Detail Settings

11.1 PharmaCode

Keywords: #PharmaCode

PharmaCode register F8 contains a number of settings that require detailed explanation. Below is a list of valid register values and detailed explanation.

0 = Disable PharmaCode decoding (Default)

1 = Enable PharmaCode decoding, no color bars expected; standard rules for all bars. Horizontally and vertically oriented symbols are decoded. Decoding is performed in the "normal" direction (left bars more significant than right bars for horizontal symbols; top bars more significant than bottom bars for vertical symbols).

3 = Enable PharmaCode decoding, Color bars expected; relaxed contrast rules for the three least significant bars. Horizontally and vertically oriented symbols are decoded. Decoding is performed in the "normal" direction (left bars more significant than right bars for horizontal symbols; top bars more significant than bottom bars for vertical symbols).

9 = Enable PharmaCode decoding, no color bars expected; standard rules for all bars. Horizontally and vertically oriented symbols are decoded. Decoding is performed in the "reverse" direction (right bars more significant than left bars for horizontal symbols; bottom bars more significant than top bars for vertical symbols).

11 = Enable PharmaCode decoding, Color bars expected; relaxed contrast rules for the three least significant bars. Horizontally and vertically oriented symbols are decoded. Decoding is performed in the "reverse" direction (right bars more significant than left bars for horizontal symbols; bottom bars more significant than top bars for vertical symbols).

12 Appendix: Example CRC16 C Code

The CRC16 required by Reader to Host packets (see Section 6.2) can be calculated using the following sample C code. This CRC16 consists of two consecutive bytes, each in range [0,255] most significant byte first. A CRC16 is calculated on each packet byte, over the entire packet, excluding the *prefix* and the CRC16 itself.

```
crc_t crc = 0;

<send firstByte>

crc = crc(crc, firstByte, firstByteSize);

<send secondByte>

crc = crc(crc, secondByte, secondByteSize)

<…>

<send crcHighByte>
<send crcLowByte>
```

```
/* crc16.h */

#ifndef     crc16_h
    #define crc16_h

    #include <stdint.h>
    #include <stddef.h>

    #ifdef __cplusplus
        extern "C" {
    #endif

    typedef uint16_t crc_t;

    crc_t crc
        ( crc_t                initialCrc
        , const unsigned char* bufPtr
        , size_t               length
        );

    #ifdef __cplusplus
        } // extern "C"
    #endif

#endif
```

```
/* crc16.c */
```

```
#include <crc16.h>

crc_t crc
    ( crc_t                initialCrc
    , const unsigned char* p
    , size_t               n
    )
{
      enum
      {
          crcBits  = 16,
          charBits = 8,
          diffBits = crcBits - charBits
      };

      crc_t c = initialCrc;

      #include "crc16tab.h"

      while( n-- )
            c = (c << charBits) ^ crcTab[( c >> diffBits ) ^ *p++];

      return c;
}

/*eof*/
```

```
/* crc16tab.h
 * crc16 table of partial remainders generated by
 * mkcrctab.c with polynomial 1021.
 * included only from within crc() function in file crc16.c
 */

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

/*eof*/
```

13 Appendix: Example CRC14 C Code

The CRC14 required by Host to Reader packets (see Section 7.2) can be calculated using the following sample C code. This CRC14 consists of two consecutive bytes, each in range [0,127] most significant byte first. A CRC16 is calculated on each packet byte, over the entire packet, excluding the *prefix* and the CRC16 itself. Bitwise AND each byte of the CRC16 checksum with 0x7F to generate the two bytes of the CRC14 checksum.

```
crc_t crc = 0;

<send firstByte>

crc = crc(crc, firstByte, firstByteSize);

<send secondByte>

crc = crc(crc, secondByte, secondByteSize)

<…>

crcHighByte = (crc >> 8) & 0x7f;
crcLowByte  =  crc       & 0x7f;

<send crcHighByte>
<send crcLowByte>
```

The CRC16 is calculated in Appendix: Example CRC16 C Code above.

14 Appendix: Custom Keyboard XML File Formatting

Keywords: #KeyboardMap

To support any USB keyboard the following xml file format is implemented.

Header “<?xml version="1.0" encoding="UTF-8"?>”
Tags:
<charatcterMapping>
Attributes:
“id” identifies the default code set used for mapping to scancodes.
“version” (optional) user supplied version of the mapping.
Data: *none*
Example: `<characterMapping id="ascii" version="1">`
<Copyright>
Attributes:
“value” (optional) user supplied copyright information.
Data: *none*
Example: <Copyright value="Copyright (c) 2013 The Code Corporation."/>
<history>
Attributes: *none*
Data: *modified tags*
Example: <history><modified version="1" date="2013">*message*</modified></history>
<modified>
Attributes:
“version”
“date”
Data: *string*
Example: <modified version="1" date="2013-04-19">*message*</modified>
<Language>
Attributes:
“name”
“id”
“revision”
“author”
“initials”
Data: *Input tags*
<Input>
Attributes:
“encoding” (not used)
“value”
“printable” (Optional printable character)
“action” (not used)

Data: *Scancodes tags*

Example: <**Input** encoding="ASCII" **value**="0000" **printable**=" " **action**="none">

<Scancodes>

Attributes:

“condition” (not used)

Data: *Scancode tags*

Example: <**Scancodes** **condition**="none">

<Scancode>

Attributes:

“value”

“modifier”

Data: *none*

Example: <**Scancode** **value**="00" **modifier**="04"/>

Note : There is a maximum scancode limit of 10 Scancode tags per Scancodes tag

Example *Language.xml* :

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE characterMapping SYSTEM "../CharacterMapping.dtd">
<characterMapping id="ascii" version="1">
    <Copyright value="Copyright (c) 2013 The Code Corporation."/>
    <history>
        <modified version="1" date="2013-04-19">Original table.</modified>
        <modified version="2" date="2013-04-26">Removed inputs above
0xFF</modified>
    </history>
    <Language name="English" id="00" revision="0.1" author="Mark Alan Ashby"
initials="MAA">
        <Input encoding="ASCII" value="0000" printable=" " action="none">
            <Scancodes condition="none">
                <Scancode value="00" modifier="04"/>
                <Scancode value="62" modifier="04"/>
                <Scancode value="00" modifier="04"/>
                <Scancode value="00" modifier="00"/>
            </Scancodes>
        </Input>
Repeat <Input><Scancodes><Scancode></Scancode></Scancodes></Input>
    </Language>
</characterMapping>
```

All technical information, specifications, costs, schedules, and all related materials quoted, expressed, or implied in this document are results of our judgment at this point in time and are only estimates based upon the information available to us. We reserve the right to modify such information as we find necessary based on client requests, available technology, and other eventualities.