

KDC200 PROGRAMMING MANUAL

Rev2.66



KoamTac
A New Wave in Auto ID

Copyright, License and Warning page

Copyright® 2008 by KoamTac, Inc. All rights reserved

No part of this publication may be reproduced or used in any form, or by any electrical or mechanical means, without permission in writing from KoamTac, Inc.

The material in this manual is subject to change without notices.

KoamTac reserves the right to make changes to any product to improve reliability, function, or design.

KoamTac doesn't assume any product liability arising out of, or in connection with, the application or use of any product, circuit, or application described herein.

Follow all warnings and instructions marked on manual and units

Use only the power source specified in this manual or marked on the units

TO PREVENT FIRE OR SHOCK HAZARD, DO NOT EXPOSE THIS UNIT TO ANY TYPE OF MOISTURE

DO NOT LOOK DIRECTLY INTO LASER or point the laser into another person's eyes. Exposure to the beam MAY CAUSE EYE DAMAGE.



CAUTION

Changes or modifications not expressly approved by the manufacturer responsible for compliance could void the user's authority to operate the equipment

WARNING

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

INFORMATION TO USER:

This equipment has been tested and found to comply with the limit of a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation; if this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

1. Reorient / Relocate the receiving antenna.
2. Increase the separation between the equipment and receiver.
3. Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
4. Consult the dealer or an experienced radio/TV technician for help

<p>CAUTION</p> <p>RISK OF EXPLOSION IF BATTERY IS REPLACED BY AN INCORRECT TYPE.</p> <p>DISPOSE USED BATTERIES ACCORDING TO THE INSTRUCTIONS</p>

Table of Contents

Internal data buffer	6
<i>Data format</i>	6
<i>How many barcodes can be stored?</i>	7
<i>What happens when this buffer is full?</i>	7
Commands Set	8
‘B’ command	11
‘bT0’ command	11
‘bT1’ command	11
‘bT2’ command	12
‘bT3’ command	12
‘bT4’ command	12
‘bT5’ command	12
‘bT6’ command	13
‘bT70’ command	13
‘bT71’ command	13
‘C’ command	14
‘c’ command	14
‘D’ command	14
‘E’ command	14
‘F’ command	15
‘GB0’ command	16
‘GB1’ command	16
‘Gb0’ command	16
‘Gb1’ command	16
‘GM’ command	17
‘GMS’ command	17
‘GMT’ command	17
‘GTS’ command	18
‘GTG’ command	18
‘H’ command	19
‘h’ command	19
‘L’ command	20
‘I’ command	20
‘M’ command	20
‘N’ command	20
‘O’ command	21
‘o’ command	21
‘P’ command	22
‘p’ command	23
‘S’ command	24
‘s’ command	24
‘T’ command	25
‘t’ command	25
‘U’ command	25

- 'u' command 26
- 'V' command 26
- 'W' command 26
- 'w' command 27
- 'Z' command 28
- 'z' command 28
- Wedging Data Format and Handshake Mode 29
 - Barcode only format 29
 - Packet data format and disabled Handshake mode 29
 - Packet data format and enabled Handshake mode 29
- Miscellaneous 30
 - KDC200 sends "?" character 30
 - KDC200 sends one to three leading "NULL" bytes with barcode payload or packet 30
- Contact Information 31

KoamTac KDC200 provides easy and rich commands set to application developers who wish to control the KDC200 and resulting data for their application.

Internal data buffer

KDC200 has 200K bytes flash memory where it stores the read barcodes. Its capacity, in terms of number of barcodes, depends on the size of barcodes. Barcodes are stored sequentially as they are scanned in the internal flash memory. The maximum number of stored barcodes is 10,240 with 200K byte limitation.

Data format

KDC200 stores the read barcodes in the internal flash memory in the following data format:

C0	Y0	D0	T0	C1	Y1	D1	T1	Cn	Yn	Dn	Tn
----	----	----	----	----	----	----	----	---	---	---	---	----	----	----	----

Where

C0, C1, ..., Cn : Total number of each barcode data (1 byte)

Y0, Y1, ..., Yn : Type of each barcode(1 byte)

D0, D1, ..., Dn : Actual barcode data of each barcode (variable size)

T0, T1, ..., Tn : Timestamp of each barcode (4 bytes)

Barcodes are stored sequentially as they are scanned in the internal flash memory. Since barcodes can have different lengths, we need to keep track of length information as well. For each barcode we first store the length of barcode plus 1 (1 byte), then the barcode type (1 byte), and then the barcode value (without the string terminator '\0').

The timestamp field has 6 sub-fields as follows:

MSB

LSB

Years (6 bits)	Months (4 bits)	Days (5 bits)	AM/PM (1 bit)	Hours (4 bits)	Minutes (6 bits)	Seconds (6 bits)
-------------------	--------------------	------------------	------------------	-------------------	---------------------	---------------------

Note:

- (1) The base year is 2000. It means the year is 2000 if Years field is 0.
- (2) The Hours range is 0 – 11 and AM/PM bit 0 means AM, and 1 means PM

How many barcodes can be stored?

The number of barcodes that can be stored in these bytes depends on the size of the barcodes

Example: If only UPCA barcodes are scanned and the check digit is not transmitted, then each barcode takes up 11 (barcode data) + 2 (length and type) + 4 (time stamp) = 17 bytes. The maximum number of UPCA barcodes that can be saved is $204,800/17 = 12,047$. However, the maximum number of barcodes can be stored is 10,240. Therefore, KDC200 can store 10,240 UPCA barcodes.

If different length barcodes are mixed, user cannot compute a priori what the maximum number of barcodes is.

What happens when this buffer is full?

When the internal flash memory is full, KDC200 displays buffer full message and ignores the command to scan barcodes.

Commands Set

KDC200 has various commands and they are quite simple. They are shown in the following table.

- User should transmit the command string within 2 seconds to avoid timeout and undesired output.
- It is recommended to insert proper delay (about 100msec) between characters when transmitting the command string.
- It is recommended to send “W” command first to wake up KDC200 before sending individual commands.

Command	Input	Output	Description	
B	None	<i>value @</i>	Returns current battery capacity(<i>value</i> = 0~100)	
b	T0	None	<i>value @</i>	Returns current settings of Bluetooth options
	T1	0	@	Turns off Bluetooth power
		1	@	Turns on Bluetooth power
	T2	None	@ or !	Start Bluetooth pairing mode
	T3	0	@ or !	Disable auto connection mode of Bluetooth at power on
		1	@ or !	Enable auto connection mode of Bluetooth at power on
	T4	0	@	Disable Bluetooth auto power on function
		1	@	Enable Bluetooth auto power on function
	T5	0	@	Disable Bluetooth auto power off function
		1	@	Enable Bluetooth auto power off function
	T6	0	@	Disable beep warning when Bluetooth auto power off function is disabled
		1	@	Enable beep warning when Bluetooth auto power off function is disabled
	T7	0	<i>value @</i>	Returns current Bluetooth auto power off timeout
1 <i>value#</i>		@	Set Bluetooth auto power off timeout	
C	<i>YMDHmS</i>	@	Setting new date and time of KDC200	
c	None	<i>YMDHmS @</i>	Returns current date and time of KDC200	
D	None	@ or !	Reading barcode (Software trigger)	
E	None	@	Erase all stored barcodes	
F	None	@	Set internal options to factory default setting	

G	B0	None	@	Unlock buttons for normal operation	
	B1	None	@	Lock buttons to prevent undesired accidental operation	
	b0	None	@	Disable beep sound	
	b1	None	@	Enable beep sound	
	M	<value# ; value#>#	@ or !		Set start display position of message from Host
		S<value>#	@ or !		Set message display duration from Host
		T<value><CR>	@		Display message from Host on KDC
	TS	value#	@ or !	Set termination character after barcode data if data format is barcode only	
TG	None	value @	Gets termination character setting after barcode data if data format is barcode only		
H	value #	@	Enable/Disable handshake while transferring packet data		
h	None	value @	Get handshake state		
L	value #	@ or !	Set minimum barcode length		
l	None	value @	Get minimum barcode length		
M	None	value @	Get serial number of KDC		
N	None	value @	Get number of barcodes stored		
O	value #	@ or !	Set scan options and flags		
o	None	value @	Get scan options and flags		
P	None	Data	Upload all stored barcode data		
p	value #	Data	Upload N _{th} stored barcode		
S	value #	@ or !	Set decoding symbologies		
s	None	value @	Get current decoding symbologies		
T	value #	@ or !	Set decoding timeout		
t	None	value @	Get current barcode read timeout		
U	value #	@ or !	Set data process mode (Wedge, Store)		
u	None	value @	Get barcode handling mode selected by command 'U'		
V	None	string @	Get firmware version		
W	None	@	Wake up KDC200 from sleep mode		
w	value #	@ or !	Select wedging data format (Barcode only, packet data)		

<i>Z</i>	<i>value #</i>	@ or !	Set security level
<i>z</i>	None	<i>value @</i>	Get current security level

‘B’ command

Parameters: None

Format: ‘B’

Output: ‘<value>@’ where <value> is the capacity of battery (0% to 100%) in hexadecimal and @ means the end of data

Purpose: Read the current battery capacity of KDC200

‘bT0’ command

Parameters: None

Format: ‘bT0’

Output: ‘<value>@’ where <value> is the current selected Bluetooth options in hexadecimal

. bit 0 : Not used

. bit 1 : Bluetooth module power is on(1)/off(0)

. bit 2 : Not used

. bit 3 : Auto connection mode is enabled(1)/disabled(0)

. bit 4 : Auto power on is enabled(1)/disabled(0)

. bit 5 : Auto power off is enabled(1)/disabled(0)

. bit 6: Beep warning is enabled(1)/disabled(0)

Purpose: Gets current Bluetooth options

‘bT1’ command

Parameters: 0 or 1 in hexadecimal (0x30 or 0x31)

Format: ‘bT10’ or ‘bT11’

Output: ‘@’ all time

Purpose: Turns on(1)/off(0) KDC200 Bluetooth module power

‘bT2’ command

Parameters: None

Format: ‘bT2’

Output: ‘@’ if successful pairing. ‘!’ if pairing is failed within 90 seconds

Purpose: KDC200 starts pairing mode

‘bT3’ command

Parameters: 0 or 1 in hexadecimal (0x30 or 0x31)

Format: ‘bT30’ or ‘bT31’

Output: ‘@’ if successful. ‘!’ if failed. This command is failed if sent when KDC200 is connected to HOST

Purpose: Enables(1)/Disables(0) KDC200 to try connection to Bluetooth HOST automatically after power on

‘bT4’ command

Parameters: 0 or 1 in hexadecimal (0x30 or 0x31)

Format: ‘bT40’ or ‘bT41’

Output: ‘@’ all time

Purpose: Enables(1)/Disables(0) KDC200 to turns on Bluetooth module power when scan button is pressed

‘bT5’ command

Parameters: 0 or 1 in hexadecimal (0x30 or 0x31)

Format: ‘bT50’ or ‘bT51’

Output: ‘@’ all time

Purpose: Enables(1)/Disables(0) KDC200 to turn Bluetooth module power off when KDC200 is disconnected from Bluetooth host for a period of time defined in ‘bT71’ command. The default power off time is 5 minute.

‘bT6’ command

Parameters: 0 or 1 in hexadecimal (0x30 or 0x31)

Format: ‘bT60’ or ‘bT61’

Output: ‘@’ all time

Purpose: Enables(1)/Disables(0) KDC200 to beep on every 15 minutes when Bluetooth power is on and Bluetooth is disconnected, but Auto power off option is disabled. KDC200 beeps 5 times with short intervals

‘bT70’ command

Parameters: None

Format: ‘bT70’

Output: ‘*value*@’ where *value* is the timeout value of Bluetooth module auto power off

Purpose: Getting current Bluetooth auto power off timeout value

‘bT71’ command

Parameters: *value*

Format: ‘bT71<*value*>#’ where *value* is the timeout value of Bluetooth module auto power off in hexadecimal

Output: ‘@’ all time

Purpose: Setting current Bluetooth auto power off timeout value from 1 minute to 30 minutes.

It will be set to 5 minute if *value* is 0; set to 30 minute if *value* is greater than 30

‘C’ command

Parameters: Date and Time value to be set in KDC200

Format: ‘CYMDHmS’ where Y is year(0 means 2000), M is month(1-12), D is day(1-31), H is hour(0-23), m is minute(0-59) an S is second(0-59)

Output: ‘@’ all the times

Purpose: Set Date and Time of KDC200

‘c’ command

Parameters: None

Format: ‘c’

Output: ‘YMDHmS@’ where Y is year(0 means 2000), M is month(1-12), D is day(1-31), H is hour(0-23), m is minute(0-59) an S is second(0-59)

Purpose: Read current Date and Time of KDC200

‘D’ command

Parameters: None

Format: ‘D’

Output: ‘!’ – KDC200 internal flash memory is full or Reading is failed

‘@’ – Barcode reading success

Purpose: Software trigger. It starts barcode scanning process. Scanning stops once a barcode is decoded or predefined scanning timeout occurs

‘E’ command

Parameters: None

Format: ‘E’

Output: ‘@’ – all the times

Purpose: Erases the internal flash memory completely. All stored data will be lost.

‘F’ command

Parameters: None

Format: ‘F’

Output: ‘@’ – all times

Purpose: Set KDC200 internal options and flags to the factory default setting

[Note] The factory default settings are:

- All symbologies are enabled
- All options are disabled except
 - UPCE_ReturnCheckDigit
 - UPCA_ReturnCheckDigit
 - EAN8_ReturnCheckDigit
 - EAN13_ReturnCheckDigit
- Minimum barcode length is set to 4
- Time-out is set to 2 seconds
- Security Level is set to 2
- Select wide scan angle
- Sleep timeout is set to 5 seconds
- Wedge and Store. Send the scanned barcode to HOST and save into flash
- Data format is barcode only.

‘GB0’ command

Parameters: None

Format: ‘GB0’

Output: ‘@’ – all times

Purpose: Unlock KDC200 up, down and scan buttons for normal operation

‘GB1’ command

Parameters: None

Format: ‘GB1’

Output: ‘@’ – all times

Purpose: Lock KDC200 up, down and scan buttons. Locking buttons prevent undesired accidental operation and keep KDC200 in sleep mode (battery save mode)

‘Gb0’ command

Parameters: None

Format: ‘Gb0’

Output: ‘@’ – all times

Purpose: Disable KDC200 beep sound

‘Gb1’ command

Parameters: None

Format: ‘Gb1’

Output: ‘@’ – all times

Purpose: Enable KDC200 beep sound

‘GM’ command

Parameters: row # and column #

Format: ‘GM<row#>;<column#>’, where $0 \leq \text{row\#} \leq 3$ and $0 \leq \text{column\#} \leq 12$ in hexadecimal

Output: ‘!’ – Invalid parameter

‘@’ – Set start position successfully

Purpose: Set start position of message

[Note]

- Default position is the 1st row and the 1st column
- KDC maintains start position value until device is reset

‘GMS’ command

Parameters: Message display duration in seconds

Format: ‘GMS<value>’, where $1 \leq \text{value} \leq 60$ in hexadecimal

Output: ‘!’ – Invalid parameter

‘@’ – Set start position successfully

Purpose: Set start position of display

[Note]

- Set the message display duration
- Default display duration is system sleep time
- KDC maintains the display duration until device is reset

‘GMT’ command

Parameters: Character string to be displayed on KDC

Format: ‘GMT<message><CR>’, where $1 \leq \text{message length} \leq 52$ in hexadecimal and <CR> is ASCII13 (0D in hex)

Output: ‘@’ – all times

Purpose: Display message on KDC

[Note]

- KDC returns @ upon receiving 52 characters
- Line would be wrapped automatically if character per line is greater than 13

‘GTS’ command

Parameters: 0, 1, 2, 3 or 4 in hexadecimal (0x30, 0x31, 0x32 or 0x33)

Format: ‘GTS<value>#’, where 0 = None, 1 = CR, 2 = LF, 3 = CR/LF, 4 = Tab

Output: ‘!’ – Invalid parameter

‘@’ – Set termination character successfully

Purpose: Set termination character after barcode data if data format is barcode only

‘GTG’ command

Parameters: None

Format: ‘GTG’

Output: ‘<value>@’, where 0 = None, 1 = CR, 2 = LF, 3 = CR/LF, 4 = Tab

value is 0, 1, 2, 3 or 4 in hexadecimal (0x30, 0x31, 0x32 or 0x33)

Purpose: Get termination character setting after barcode data if data format is barcode only

‘H’ command

Parameters: 0 or 1 in hexadecimal (0x30 or 0x31)

Format: “H<value>#”, where *value* = 0 disables handshake, 1 enables handshake
value is 0 or 1 in hexadecimal (0x30, 0x31)

Output: ‘!’ – invalid parameter
 ‘@’ – all other times

Purpose: Enable/Disable handshake with HOST while sending packet data. KD100 doesn’t perform handshake if data format is barcode only.

[Note]

- Handshake mode is applicable only ‘p’ command and wedging in packet data format only.
- Handshake mode is NOT applicable to ‘P’ command and wedging in data only format
- If handshake mode is enabled,
 - KDC200 expects HOST to send either success or failure response to KDC200 once HOST receives packet data from KDC200.
 - HOST is required to send a success response if it receives a packet data successfully by sending a character ‘@’. If HOST detected an error while receiving a packet data from KDC200, HOST should send a failure response to KDC200 by sending a character ‘!’.
 - KDC200 tries multiple transmission up to 10 times if KDC200 doesn’t get response within 1 second or receives failure response ‘!’ from HOST.

‘h’ command

Parameters: None

Format: ‘h’

Output: ‘<value>@’ where 0 means disabled, 1 means enabled
value is 0 or 1 in hexadecimal (0x30, 0x31)

Purpose: Get the current setting of handshaking mode. 0 means disabled and 1 means enabled

‘L’ command

Parameters: Minimum length of barcode between 2 to 36 in hexadecimal

Format: “L<value>#” or “l<value>#” where <value> is the value of parameter in hexadecimal

Output: ‘!’ – invalid parameter

‘@’ – all other times

Purpose: Set the minimum length of barcodes

[Example] To set the minimum barcode length to 20, send command “L14#”.

“L4#” sets the minimum length to 4.

‘l’ command

Parameters: None

Format: ‘l’

Output: ‘<value>@’ where <value> is the minimum length of barcodes in hexadecimal and @ means the end of data

Purpose: Get the current setting of minimum barcode length

‘M’ command

Parameters: None

Format: ‘M’

Output: ‘yymmxxxxx@’ where yy is manufacture year and mm is month, xxxxxx is the serial number and @ means the end of data

Purpose: Get the serial number of KDC200

‘N’ command

Parameters: None

Format: ‘N’

Output: ‘<value>@’ where <value> is the number of stored barcodes in hexadecimal and @ means the end of data

Purpose: Get the number of stored barcodes in KDC200 internal flash memory

‘O’ command

Parameters: The options (it is unsigned 32-bit number)

Format: ‘O<value>#’ where <value> is the value of parameter in hexadecimal

Output: ‘!’ – invalid parameter

‘@’ – all other times

Purpose: Enables the selected options

[Example] Sending “O0#” disables all the options

[Note] Set the appropriate bit to “1” to enable an option

Option	bit#	Option	bit#
CodaBar_NoStartStopChars	0	UPCA_as_EAN13	19
ReverseDirection	4	I2of5_VerifyCheckDigit	22
UPCE_as_UPCA	9	Code39_VerifyCheckDigit	23
EAN8_as_EAN13	10	I2of5_ReturnCheckDigit	26
UPCE_as_EAN13	11	Code39_ReturnCheckDigit	27
ReturnCheckDigit	12	UPCE_ReturnCheckDigit	28
VerifyCheckDigit	13	UPCA_ReturnCheckDigit	29
WideScanAngle	14	EAN8_ReturnCheckDigit	30
HighFilterMode	15	EAN13_ReturnCheckDigit	31

‘o’ command

Parameters: None

Format: ‘o’

Output: ‘<value>@’ where <value> is the selected barcode options in hexadecimal and @ means the end of data

Purpose: Get the barcode options saved in KDC200. Please see the table of command ‘O’ for the detailed explanation of options

'P' command

Parameters: None.

Format: 'P'

Output: A multi-byte string (see below)

Purpose: Get all barcodes data stored in KDC200 internal flash memory. 'P' command does **not** erase the stored barcodes.

[Note] The format of the output is as follows:

N	C ₀	Y ₀	D ₀	T ₀	C ₁	Y ₁	D ₁	T ₁	C _n	Y _n	D _n	T _n
---	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	---	---	---	---	---	---	---	---	---	---	---	---	---	----------------	----------------	----------------	----------------

Where

N : Total number of bytes to be sent(3 bytes)

C₀, C₁, ..., C_n : Total number of each barcode data(1 byte) including Y/D/T bytes (not including C byte)

Y₀, Y₁, ..., Y_n : Type of each barcode(1 byte)

D₀, D₁, ..., D_n : Actual barcode data of each barcode(variable size)

T₀, T₁, ..., T_n : Timestamp of each barcode(4 bytes)

- The first 3 bytes specify the total number of bytes being sent. Then the entire populated contents of the internal flash memory are sent as is.
- After the 3rd byte, the message can be thought of in terms of variable size blocks with each block of bytes representing information on a single barcode. These blocks are placed next to each other with no "empty" bytes between them.
- 'P' command does not support Handshake mode

[Example] The output of "P" command is "00 00 15 11 02 30 31 32 35 34 36 36 31 32 32 39 36 00 42 17 24"

N: 00 00 15 → 21 bytes

C₀: 11 → 17 bytes

Y₀: 02 → UPCA

D₀: 30 31 32 35 34 36 36 31 32 32 39 36 → 012546612296 UPCA barcode

T₀: 00 42 17 24 →

Years	Months	Days	AM/PM	Hours	Minutes	Seconds
000000	0001	00001	0	0001	011100	100100
2000	1	1	AM	1 hr	28 min	36 sec

‘p’ command

Parameters: The number which indicates the position of barcode data to be sent

Format: ‘p<value>#’ where <value> is the number of barcode data stored in the internal flash to upload in hexadecimal. The first stored barcode location is “00”, not “01”

Output: A multi-byte string (see below). KDC200 sends only one barcode data at one time

Purpose: Get the Nth barcodes data stored in KDC200 internal flash memory. ‘p’ command does not erase the barcode in the internal flash

[Note]

S	N	C	Y	D	T	K
---	---	---	---	---	---	---

Where:

S: Start byte 0x03(1 byte)

N: Total number of bytes to be sent(3 bytes)

C: Total number of each barcode data(1 byte)

Y: Type of each barcode(1 byte)

D: Actual barcode data of each barcode(variable size)

T: Timestamp of each barcode(4 bytes)

K: Check sum byte(1 byte). All sum except S should be 0

- To upload all stored barcode data by using this command, it is required to call this command N times
- User should send “p<value>#” string within 2 seconds
 - Transmit time out is extended to 10 seconds from Firmware version 1.63(2.63)
- KDC200 tries multiple transmission up to 10 times if Handshake mode is enabled and KDC200 doesn’t get response from HOST within 1 second
- p command will output timeout result if user fails to send “p<value>#” string within 2 seconds
 - Firmware version 1.00(2.00) ~ 1.62(2.62): returns the first stored barcode data
 - Firmware version 1.63(2.63) ~ current: returns “!”, invalid parameter

[Example] Sending “p14#” string to KDC to upload the barcode data stored in 20th position

‘S’ command

Parameters: The symbology selection (it is unsigned 32-bit number)

Format: ‘S<value>#’ where <value> is the value of parameter in hexadecimal (from 1 to 4 bytes)

Output: ‘!’ – invalid parameter

‘@’ – all other times

Purpose: Enables the selected symbologies

[Example] Sending “SFFFFFFFF#” enables all the symbologies

[Note] Set the appropriate bit to “1” to select a symbology

Symbology	bit#	Symbology	bit#
EAN13	0	EAN128	9
EAN8	1	Code93	10
UPCA	2	Code35	11
UPCE	3	BooklandEAN	12
Code39	4	EAN13withAddon	13
ITF14	5	EAN8withAddon	14
Code128	6	UPCAwithAddon	15
I2of5	7	UPCEwithAddon	16
Codabar	8	PDF417	31

‘s’ command

Parameters: None

Format: ‘s’

Output: ‘xxxxxxx@’ where xxxxxxx is the selected symbologies and @ means the end of data

Purpose: Get the selected symbologies. Please see the table of command ‘S’ for detailed explanation of each symbology

‘T’ command

Parameters: Time in msec. Valid range is 100msec to 10sec.

Format: “T<value>#” where <value> is the value of parameter in hexadecimal (from 1 to 4 bytes)

Output: ‘!’ – invalid parameter

‘@’ – all other times

Purpose: Set the timeout for barcode read operation

[Example] Sending “T3e8#” sets the timeout to 1sec.

‘t’ command

Parameters: None

Format: ‘t’

Output: ‘<value>@’ where <value> is the timeout value for barcode read operation in hexadecimal and @ means the end of data

Purpose: Get the current read timeout setting

‘U’ command

Parameters: 0, 1,2 or 3

Format: “U<value>#” where <value> is the value of parameter in hexadecimal

- 0: KDC200 doesn’t store scanned barcode in the memory and just transmits it to the host
- 1: KDC200 stores scanned data in the memory and transmits it to the host
- 2: KDC200 stores scanned data in the memory but doesn’t transmit it to the host
- 3: KDC200 stores scanned data in the memory only if it succeeds to send read barcode to host
- 4: KDC200 stores scanned date in the memory only if it fails to send read barcode to host

Output: ‘!’ – invalid parameter

‘@’ – all other times

Purpose: Set the data process mode of KDC200

‘u’ command

Parameters: None

Format: ‘u’

Output: ‘<value>@’ where <value> is the value of data process mode

- 0: KDC200 doesn’t store scanned barcode in the memory and just transmits it to the host
- 1: KDC200 stores scanned data in the memory and transmits it to the host
- 2: KDC200 stores scanned data in the memory but doesn’t transmit it to the host
- 3: KDC200 stores scanned data in the memory only if it succeeds to send read barcode to host
- 4: KDC200 stores scanned data in the memory only if it fails to send read barcode to host

Purpose: Get the current setting of barcode handling mode

‘V’ command

Parameters: None

Format: “V”

Output: ‘<string>@’ where <string> is KDC200 firmware version and @ means the end of data

Purpose: Get the firmware version of KDC200

‘W’ command

Parameters: None

Format: “W”

Output: ‘@’ – all times

Purpose: Wake up KDC200 from sleep mode

[Note]

- KDC200 wouldn’t respond to “W” command if it is in sleep mode.
- It is required to send ‘W’ command multiple times until receiving ‘@’ output

‘w’ command

Parameters: 0 or 1

Format: ‘w<value>#’ where <value> is wedging format in hexadecimal

- 0: KDC200 sends barcode only to HOST,
- 1: KDC200 sends PACKET data and wait for response(‘@’) from HOST

Output: ‘!’ – invalid parameter

‘@’ – all other times

Purpose: KDC200 transmits scanned data to HOST if Wedge mode is enabled. User can set Wedge mode using ‘U’ command. KDC200 can send barcode data only or PACKET data

[Note]

- KDC200 tries multiple transmission up to 10 times if Handshake mode is enabled and KDC200 doesn’t get response from HOST within 1 second
- KDC200 supports two kind of data format when sending data to HOST

– Barcode only

<i>Barcode data</i>	<i>Termination Character</i>
---------------------	------------------------------

where: Barcode data is the actual read barcode data and termination characters are “None”, “CR (0x0d, ‘r’)”, “LF (0x0a, ‘n’)” or “CR + LF”

– PACKET DATA

S	N	C	Y	D	T	K
---	---	---	---	---	---	---

where:

S : Start byte 0x03(1 byte)

N : Total number of bytes to be sent(3 bytes)

C : Total number of each barcode data(1 byte) including Y/D/T/K bytes (not including C byte)

Y : Type of each barcode(1 byte)

D : Actual barcode data of each barcode(variable size)

T : Timestamp of each barcode(4 bytes)

K : Check sum byte(1 byte). All sum except S should be 0.

'Z' command

Parameters: Security level (valid values: 1 to 4)

Format: "Z<value>#" where <value> is the value of parameter in hexadecimal.

1 is the lowest security level and 4 is the highest security level.

Output: '!' – invalid parameter

'@' – all other times

Purpose: Set the security level

'z' command

Parameters: None

Format: 'z'

Output: '<value>@' where <value> is the current security level in hexadecimal and @ means the end of data. 1 is the lowest security level and 4 is the highest security level.

Purpose: Get the current security level value

Wedging Data Format and Handshake Mode

KDC200 supports user selectable two data formats in Wedging mode:

- Barcode only and
- Packet data

Barcode only format

Barcode only data format doesn't support handshake mode.

Packet data format and disabled Handshake mode

- KDC200 does not perform handshaking if Wedging data format is packet data but handshake mode is disabled.

Packet data format and enabled Handshake mode

- KDC200 expects the HOST to send either success or failure response to KDC200 once HOST received packet data from KDC200 if Wedging data format is packet data and handshake mode is enabled.
- HOST is required to send a success response if it receives a packet data successfully by sending a character '@'. If HOST detected an error while receiving a packet data from KDC200, HOST should send a failure response to KDC200 by sending a character '!'.
If HOST sends a success response to KDC200, KDC200 will send a success response to HOST by sending a character '@'.
- KDC200 tries multiple transmission up to 10 times if Handshake mode is enabled and KDC200 doesn't get response from HOST within 1 second

Miscellaneous

KDC200 sends "?" character

“?” is sent by KDC200 when the device receives undefined character (command) from Host. Application ignores“?” character

KDC200 sends one to three leading “NULL” bytes with barcode payload or packet

KDC200 Bluetooth module goes to sleep mode if not used for a while. KDC200 sends out one to three NULL bytes to wake up Bluetooth module before sending out barcode payload or packet. Application should ignore these leading “NULL” bytes.

Contact Information

KDC Technical Support Team

KoamTac, Inc.

116 Village Blvd. Suite 200

Princeton, NJ 08540

USA

TEL: +1-908-734-4335

FAX: +1-609-226-4373

Email: support@koamtac.com

URL: www.koamtac.com