

REVISION

**2.82**

**KoamTac**  
*A New Wave in Auto ID*

# Programming Manual

## KDC300



July, 2009

## Copyright, License and Warning page

**Copyright**© 2009 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

### CAUTION

RISK OF EXPLOSION IF BATTERY IS REPLACED BY AN INCORRECT TYPE.

DISPOSE USED BATTERIES ACCORDING TO THE INSTRUCTIONS

# Table of Contents

Internal data buffer .....	5
<i>Data Format</i> .....	5
<i>How many barcodes can be stored?</i> .....	7
<i>What happens when the buffer is full?</i> .....	7
Commands Set .....	8
' <i>B</i> ' (Read battery level).....	11
' <i>bT0</i> ' (Read Bluetooth options).....	12
' <i>bT1</i> ' (Bluetooth power on/off).....	13
' <i>bT2</i> ' (Bluetooth pairing Mode).....	14
' <i>bT3</i> ' (Bluetooth auto connection).....	15
' <i>bT4</i> ' (Bluetooth auto power on).....	16
' <i>bT5</i> ' (Bluetooth auto power off).....	17
' <i>bT6</i> ' (Bluetooth disconnection beep warning).....	18
' <i>bT70</i> ' (Read Bluetooth auto power off timeout value).....	19
' <i>bT71</i> ' (Set Bluetooth auto power off timeout value).....	20
' <i>bT80</i> ' (Disable Bluetooth power off message to host).....	21
' <i>bT81</i> ' (Enable Bluetooth power off message to host).....	22
' <i>bT9</i> ' (Bluetooth MAC address).....	23
' <i>bTA0</i> ' (Set KDC300 in Bluetooth Slave Mode).....	24
' <i>bTA1</i> ' (Set KDC300 in Bluetooth Master Mode).....	25
' <i>bTC</i> ' (Activates a connection to predefined Bluetooth MAC address and PIN code).....	26
' <i>bTD</i> ' (Set Bluetooth MAC address).....	27
' <i>bTE</i> ' (Set Bluetooth PIN code).....	28
' <i>C</i> ' (Set date and time).....	29
' <i>c</i> ' (Read date and time).....	30
' <i>D</i> ' (Software trigger).....	31
' <i>E</i> ' (Erase memory).....	32
' <i>e</i> ' (Erase the last stored data).....	33
' <i>F</i> ' (Factory default).....	34
' <i>GB0</i> ' (Unlock button).....	35
' <i>GB1</i> ' (Lock button).....	36
' <i>Gb0</i> ' (Beeper off).....	37
' <i>Gb1</i> ' (Beeper on).....	38
' <i>GC0</i> ' (Disable port connection status display).....	39
' <i>GC1</i> ' (Enable port connection status display).....	40
' <i>GD0</i> ' (Unlock date and time setting).....	41
' <i>GD1</i> ' (Lock date and time setting).....	42

'GM' (Message start position).....	43
'GMB0' (Failure Alert Beep).....	44
'GMB1' (Success Alert Beep).....	45
'GMS' (Message display duration).....	46
'GMT' (Display message).....	47
'GS0' (Finish Synchronization).....	48
'GS1' (Start Synchronization).....	49
GTS' (Set termination character).....	50
'GTG' (Read termination character).....	51
'GW' (Menu Password Protection).....	52
'GX0' (Disable Auto Menu Exit).....	53
'GX1' (Enable Auto Menu Exit).....	54
'GY0' (Display data, type, date and time).....	55
'GY1' (Display data, date and time, battery status).....	56
'GY2' (Display data, type, battery status).....	57
'H' (Set handshake mode).....	58
'h' (Read handshake mode).....	59
'L' (Set minimum barcode length).....	60
'l' (Read minimum barcode length).....	61
'M' (Read serial number).....	62
'N' (Get # of stored barcode).....	63
'O' (Set barcode option).....	64
'o' (Read barcode option).....	68
'P' (Download all stored barcode).....	69
'p' (Download n <sup>th</sup> stored barcode).....	71
'S' (Select symbology).....	74
's' (Read selected symbology).....	77
'T' (Set Timeout).....	78
't' (Read timeout value).....	79
'U' (Set data process mode).....	80
'u' (Read data process mode).....	81
'V' (Read firmware version).....	82
'W' (Wake up).....	83
'w' (Set data format).....	84
'Z' (Set security level).....	85
'z' (Read security level).....	86
Wedging Data Format and Handshake Mode .....	87
Barcode only format.....	87
Packet data format and disabled Handshake mode.....	87
Packet data format and enabled Handshake mode.....	87

Miscellaneous ..... 88  
    *KDC300 sends "?" character*..... 88  
    *KDC300 sends one to three leading "NULL" bytes*..... 88  
Contact Information ..... 89

KoamTac KDC300 provides easy and extensive commands set to application developers who wish to control the KDC300 and resulting data for their application.

## Internal data buffer

---

KDC300 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

User can distinguish Normal mode, Master/Slave Application and user created Application records by checking the second last byte in the record. It is user created Application record if the second last byte is 0xFF in hexadecimal (11111111 in binary). It is either Normal mode or Master/Slave Application records if the second last byte is NOT 0xFF. User can further distinguish Normal mode and Master/Slave Application records by checking the first two bits of barcode type byte defined in the below.

#### **Normal mode and Master/Slave Application**

KDC300 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 (2 bytes)

- **2 bytes for KDC200P/KDC300, 1 byte for KDC100/200**

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

- The 1<sup>st</sup> 2 bits distinguish Normal mode or Master/Slave Application records
  - 00: Normal barcode
  - 01: Mismatched slave barcode
  - 10: Matched slave barcode
  - 11: Master barcode
- The remaining 6 bits indicate barcode type

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

**User Applications generated by Application Generator**

KDC300 stores the read barcodes in the internal flash memory in the following data format:

C0	Y0	D0	T0	Q0	C1	Y1	D1	T1	Q1.	.....	Cn	Yn	Dn	Tn	Qn.
----	----	----	----	----	----	----	----	----	-----	-------	----	----	----	----	-----

Where

- C0, C1, ..., Cn : Total number of each barcode data (2 bytes)
  - o **2 bytes for KDC200P/KDC300, 1 byte for KDC100/200**
- Y0, Y1, ..., Yn : Type of each barcode(1 byte)
  - o The 1<sup>st</sup> 2 bits indicate record type
    - 0 : Data collected in Normal mode
    - 1 : Step 1 data collected in Application mode
    - 2 : Step 2 data collected in Application mode
    - 3 : Step 3 data collected in Application mode
  - o The remaining 6 bits indicate barcode type
- D0, D1, ..., Dn : Actual barcode data of each barcode (variable size)
- T0, T1, ..., Tn : Timestamp of each barcode (4 bytes)
- Q0, Q1, ..., Qn : (2 bytes)
  - o The 1<sup>st</sup> byte indicates Predefined or User Application
    - 0xFF : User Application

- The 2<sup>nd</sup> byte indicates Quantity
  - MSB(bit 7) is the compliant data indicator
    - 0: Compliant data
    - 1: Not compliant data
  - Remaining 7 bits indicate quantity (1~128)

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, KDC300 can store 10,240 UPCA barcodes.*

## What happens when the buffer is full?

When the internal flash memory is full, KDC300 displays “Buffer Full” message and ignores the command to scan barcodes.

## Commands Set

KDC300 has various commands and they are quite simple. They are shown in the following table.

- Command parameters are either character or hexadecimal numbers. Please read each command description and example carefully.
- 0xNN means one byte data “NN” in hexadecimal format. For example, 0x30 means hexadecimal number 30 which is equivalent to character “0”.
- 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 KDC300 before sending individual commands.

Command	Input	Output	Description	
B	None	value @	Returns current battery capacity(value = 0~100)	
b	T0	None	value @	Returns current settings of <i>Bluetooth</i> options
	T1	0	@	Turns off <i>Bluetooth</i> power
		1	@	Turns on <i>Bluetooth</i> power
	T2	None	@ or !	Start <i>Bluetooth</i> pairing mode
	T3	0	@ or !	Disable auto connection mode of <i>Bluetooth</i> at power on
		1	@ or !	Enable auto connection mode of <i>Bluetooth</i> at power on
	T4	0	@	Disable <i>Bluetooth</i> auto power on function
		1	@	Enable <i>Bluetooth</i> auto power on function
	T5	0	@	Disable <i>Bluetooth</i> auto power off function
		1	@	Enable <i>Bluetooth</i> auto power off function
	T6	0	@	Disable beep warning when <i>Bluetooth</i> auto power off function is disabled
		1	@	Enable beep warning when <i>Bluetooth</i> auto power off function is disabled
	T7	0	value @	Returns current <i>Bluetooth</i> auto power off timeout
		1 value#	@	Set <i>Bluetooth</i> auto power off timeout
	T8	0	@	Disable <i>Bluetooth</i> power off message to host
		1	BTOFF@	Enable <i>Bluetooth</i> power off message to host
	T9		value @	Returns KDC <i>Bluetooth</i> Mac address
	TA	0	@	Set <i>Bluetooth</i> device to slave mode
1		@	Set <i>Bluetooth</i> device to master mode	
TC		@	Connect to predefined <i>Bluetooth</i> MAC address and PIN code	
TD	<value>#	@ or !	Set <i>Bluetooth</i> MAC address	
TE	<value>#	@ or !	Set PIN code. No PIN code if <value> is Null.	
C	YMDHmS	@	Setting new date and time of KDC300	

c	None	YMDHmS @	Returns current date and time of KDC300	
D	None	@ or !	Reading barcode (Software trigger)	
E	None	@	Erase all stored barcodes	
e	None	@	Erase the last stored barcode	
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
	C0	None	@	Disable <i>Bluetooth</i> , Serial and USB connection status display
	C1	None	@	Enable <i>Bluetooth</i> , Serial and USB connection status display
	D0	None	@	Unlock date and time setting
	D1	None	@	Lock date and time setting
	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
		B0	@	Failure Alert Beep
		B1	@	Success Alert Beep
	S0	None	@	Start Synchronization Mode
	S1	None	@	Finish Synchronization Mode
	TS	Value#	@ or !	Set termination character after barcode data if data format is barcode only
	TG	None	value @	Get termination character setting after barcode data if data format is barcode only
	W	0	@	Disable menu password protection
		1<password>#	@	Enable menu password protection
	X	0	@	Disable automatic menu exit feature
1		@	Enable automatic menu exit feature	
Y0	None	@	Display scanned barcode data, barcode type, date and time	
Y1	None	@	Display scanned barcode data, date and time, battery status	
Y0	Note	@	Display scanned barcode data, barcode type, battery status	
H	value #	@	Enable/Disable handshake mode 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 <sub>th</sub> 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 KDC300 from sleep mode	

w	<i>value #</i>	@ or !	Select wedging data format (Barcode only, packet data)
Z	<i>value #</i>	@ or !	Set security level
z	None	<i>value @</i>	Get current security level

## 'B' (Read battery level)

Parameters: None

Format: 'B'

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

Purpose: Read the current battery capacity of KDC300

[Example] Output: 0x00 0x00 0x00 0x64 0x40

- Ignore the first three leading NULL bytes
- 64% in hexadecimal (100% in decimal)
- Hexadecimal 40 is "@" character

## 'bT0' (Read *Bluetooth* options)

Parameters: None

Format: 'bT0'

Output: '<value>@' where <value> is the current selected *Bluetooth* options in hexadecimal and @ means the end of <value>

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

[Example] Output: 0x00, 0x00, 0x00, 0x60, 0x40

- 0x60: 01100000 in binary, Bit 5, 6 are enabled. Bit 1,3,4 are disabled
- 0x40: ASCII code of character "@" in hexadecimal

'bT1' (*Bluetooth* power on/off)

Parameters: 0 or 1

Format: 'bT1&lt;value&gt;', where &lt;value&gt; is 0 or 1

value	Power	Character	Hex No.
0	Off	"0"	30
1	On	"1"	31

Output: '@'

Purpose: Turns on(1)/off(0) KDC300 *Bluetooth* power[Example] Send 4 bytes data (0x62, 0x54, 0x31, 0x30) to turn off *Bluetooth* power

- 0x62: ASCII code of character "b" in hexadecimal
- 0x54: ASCII code of character "T" in hexadecimal
- 0x31: ASCII code of character "1" in hexadecimal
- 0x30: ASCII code of character "0" in hexadecimal
- bT10

## 'bT2' (*Bluetooth* pairing Mode)

Parameters: None

Format: 'bT2'

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

Purpose: KDC300 starts pairing mode

'bT3' (*Bluetooth* auto connection)

Parameters: 0 or 1

Format: 'bT3&lt;value&gt;', where &lt;value&gt; is 0 or 1

value	Auto Connection	Character	Hex No.
0	Disable	"0"	30
1	Enable	"1"	31

Output: '@' if successful. '!' if failed. This command will be failed if sent when KDC300 is connected to the HOST

Purpose: Enables(1)/Disables(0) KDC300 to try connection to *Bluetooth* HOST automatically after power on

[Note] KDC300 auto connection mode can't be changed to "Disable" from "Enable" while KDC300 is connected to the host. One should disconnect host connection first then execute "bT30".

'bT4' (*Bluetooth* auto power on)

Parameters: 0 or 1

Format: 'bT4&lt;value&gt;', where &lt;value&gt; is 0 or 1

value	Power on	Character	Hex No.
0	Disable	"0"	30
1	Enable	"1"	31

Output: '@'

Purpose: Enables(1)/Disables(0) KDC300 to turn on *Bluetooth* module power when scan button is pressed

'bT5' (*Bluetooth* auto power off)

Parameters: 0 or 1

Format: 'bT5<value>', where <value> is 0 or 1

value	Power Off	Character	Hex No.
0	Disable	"0"	30
1	Enable	"1"	31

Output: '@'

Purpose: Enables(1)/Disables(0) KDC300 to turn *Bluetooth* module power off when KDC300 is disconnected from *Bluetooth* host for a period of time defined in 'bT71' command. The default power off time is 5 minutes.

'bT6' (*Bluetooth* disconnection beep warning)

Parameters: 0 or 1

Format: 'bT6&lt;value&gt;', where &lt;value&gt; is 0 or 1

value	Power on	Character	Hex No.
0	Disable	"0"	30
1	Enable	"1"	31

Output: '@'

Purpose: Enables(1)/Disables(0) KDC300 to beep on every 15 minutes when *Bluetooth* power is on and *Bluetooth* is disconnected, but Auto power off option is disabled. KDC300 beeps 5 times with short intervals

## 'bT70' (Read *Bluetooth* auto power off timeout value)

Parameters: None

Format: 'bT70'

Output: '<value>@', where <value> is the timeout value (minute) of *Bluetooth* auto power off in hexadecimal and @ means the end of <value>

Purpose: To get current *Bluetooth* auto power off timeout value

[Example] Output: 0x00, 0x00, 0x00, 0x05, 0x40

- 0x05: *Bluetooth* auto power off is 5 minutes
- 0x40: ASCII code of character "@" in hexadecimal

'bT71' (Set *Bluetooth* auto power off timeout value)

Parameters: Timeout value of *Bluetooth* auto power off

Format: 'bT71<value>#' where *value* is between 1 and 30 minutes

Minimum Length	Character	Hex. Number
1~9	"1" ~ "9"	31 ~ 39
10~15	"A" ~ "F"	41 ~ 46
16~25	"1","0" ~ "1","9"	31,30 ~ 31,39
26~30	"1","A" ~ "1","D"	31,41 ~ 31,44

Output: '@'

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

[Note] It will be set to 5 minutes if *value* is 0; set to 30 minutes if *value* is greater than 30

[Example] Send 7 bytes data (0x62, 0x54, 0x37, 0x31, 0x31, 0x31, 0x23) to set *Bluetooth* auto power off timeout to 17 (0x11) minutes

- 0x62: ASCII code of character "b" in hexadecimal
- 0x54: ASCII code of character "T" in hexadecimal
- 0x37: ASCII code of character "7" in hexadecimal
- 0x31: ASCII code of character "1" in hexadecimal
- 0x31: ASCII code of character "1" in hexadecimal
- 0x31: ASCII code of character "1" in hexadecimal
- 0x23: ASCII code of character "#" in hexadecimal
- bT7111#

## 'bT80' (Disable *Bluetooth* power off message to host)

Parameters: None

Format: 'bT80'

Output: '@'

Purpose: KDC300 can send *Bluetooth* power off message "BTOFF" to host upon executing "bT10" *Bluetooth* power off command. KDC300 wouldn't send this "BTOFF" message unless user enables *Bluetooth* power off message using "bT81" command. The default is disable *Bluetooth* power off message.

## 'bT81' (Enable *Bluetooth* power off message to host)

Parameters: None

Format: 'bT81'

Output: 'BTOFF@'

Purpose: KDC300 can send *Bluetooth* power off message "BTOFF" to host upon executing "bT10" *Bluetooth* power off command. KDC300 would send this "BTOFF" message if user enables *Bluetooth* power off message using "bT81" command. The default is disable *Bluetooth* power off message.

'bT9' (*Bluetooth* MAC address)

Parameters: None

Format: 'bT9'

Output: '<value>@', where <value> is the MAC address of KDC300 *Bluetooth* module and @ means the end of <value>

Purpose: To get current *Bluetooth* MAC address of KDC300 *Bluetooth* module

## 'bTA0' (Set KDC300 in *Bluetooth* Slave Mode)

Parameters: None

Format: 'bTA0'

Output: '@'

Purpose: KDC300 can be configured to *Bluetooth* master or slave Mode. 'bTA0' command would set KDC300 to *Bluetooth* slave mode. The default is slave mode.

Note: This function is only applicable to the device after SN 4358.

## 'bTA1' (Set KDC300 in *Bluetooth* Master Mode)

Parameters: None

Format: 'bTA1'

Output: '@'

Purpose: KDC300 can be configured to *Bluetooth* master or slave Mode. 'bTA1' command would set KDC300 to *Bluetooth* master mode. The default is slave mode.

Note: This function is only applicable to the device after SN 4358.

'bTC' (Activates a connection to predefined *Bluetooth* MAC address and PIN code)

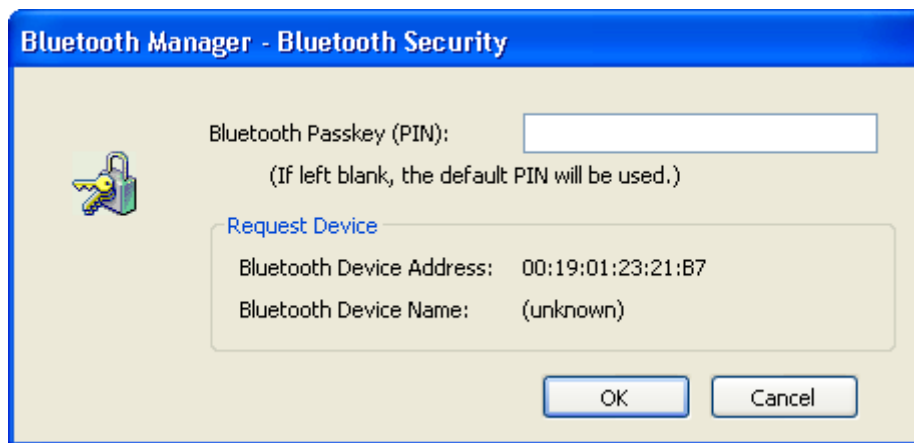
Parameters: None

Format: 'bTC'

Output: '@'

Purpose: User can pair and connect to other *Bluetooth* device using predefined *Bluetooth* MAC address and PIN code. User may have to set KDC300 as master mode using "bTA" command depends on corresponding device *Bluetooth* stack.

Note: Please enter PIN code "0000" if host Bluetooth manager requests to enter KDC300 Bluetooth PIN code.



Note: This function is only applicable to the device after SN 4358.

'bTD' (Set *Bluetooth* MAC address)

Parameters: Bluetooth MAC address of pairing device

Format: 'bTD<value>#', where <value> is 12 digits hexadecimal *Bluetooth* MAC address

Minimum Length	Character	Hex. Number
0~9	"0" ~ "9"	30 ~ 39
10~15	"A" ~ "F"	41 ~ 46

Output: '!' – invalid parameter

'@' – all other times

Purpose: User can pair and connect to other *Bluetooth* device using predefined *Bluetooth* MAC address and PIN code. User may have to set KDC300 as master mode using "bTA" command depends on corresponding device *Bluetooth* stack.

Note: This function is only applicable to the device after SN 4358.

'bTE' (Set *Bluetooth* PIN code)

Parameters: Bluetooth MAC address of pairing device

Format: 'bTE<value>#', where <value> is 1 to 7 digits *Bluetooth* PIN code

Minimum Length	Character	Hex. Number
0~9	"0" ~ "9"	30 ~ 39

Output: '!' – invalid parameter

'@' – all other times

Purpose: User can pair and connect to other *Bluetooth* device using predefined *Bluetooth* MAC address and PIN code. User may have to set KDC300 as master mode using "bTA" command depends on corresponding device *Bluetooth* stack.

Note: This function is only applicable to the device after SN 4358.

## 'C' (Set date and time)

Parameters: Date and Time value to be set in KDC300

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) and S is second(0-59). Y, M, D, H, m and S are numbers in hexadecimal.

Output: '@'

Purpose: Set Date and Time of KDC300

[Example] Send 6 bytes data (0x00, 0x02, 0x03, 0x17, 0x0C, 0x01) to set date and time to 2000/02/03:23:12:01

[Note1] KDC doesn't check the validity of date and time format. Invalid data can malfunction the device. Please check the validity of date and time format carefully.

[Note2] Year range is between 1970 and 2069.

Year	Hex No.
1970 ~ 1999	0x46 ~ 0x63
2000 ~ 2069	0x00 ~ 0x45

## 'c' (Read date and time)

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) and S is second(0-59) and @ means the end of data. Y, M, D, H, m and S are hexadecimal numbers. Year range is between 1970 and 2069.

Year	Hex No.
1970 ~ 1999	0x46 ~ 0x63
2000 ~ 2069	0x00 ~ 0x45

Purpose: Read current Date and Time of KDC300

[Example] Output: 0x00, 0x02, 0x03, 0x17, 0x0C, 0x01

- 0x00: Year 2000
- 0x02: February
- 0x03: 3<sup>rd</sup>
- 0x17: 23 Hour (Hexadecimal 17 is equal to 23 in decimal)
- 0x0C: 12 Min. (Hexadecimal C is equal to 12 in decimal)
- 0x01: 1 Sec
- 2000/02/03:23:12:01

## 'D' (Software trigger)

Parameters: None

Format: 'D'

Output: '!' – KDC300 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' (Erase memory)

Parameters: None

Format: 'E'

Output: '@'

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

## 'e' (Erase the last stored data)

Parameters: None

Format: 'e'

Output: '@'

Purpose: Erases the last stored data.

## 'F' (Factory default)

Parameters: None

Format: 'F'

Output: '@'

Purpose: To set KDC300 internal options and flag 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' (Unlock button)

Parameters: None

Format: 'GB0'

Output: '@'

Purpose: Unlock KDC300 up, down and scan buttons for normal operation

## 'GB1' (Lock button)

Parameters: None

Format: 'GB1'

Output: '@'

Purpose: To lock KDC300 up, down and scan buttons. Locking buttons prevents undesired accidental operation and keeps KDC300 in sleep mode (battery saving mode)

## 'Gb0' (Beeper off)

Parameters: None

Format: 'Gb0'

Output: '@'

Purpose: Disable KDC300 beep sound

## 'Gb1' (Beeper on)

Parameters: None

Format: 'Gb1'

Output: '@'

Purpose: To enable KDC300 beep sound

## 'GC0' (Disable port connection status display)

Parameters: None

Format: 'GC0'

Output: '@'

Purpose: Disable *Bluetooth*, Serial and USB connection status display. KDC300 does not display connected and disconnected message.

## 'GC1' (Enable port connection status display)

Parameters: None

Format: 'GC1'

Output: '@'

Purpose: Enable *Bluetooth*, Serial and USB connection status display. KDC300 displays connected and disconnected message.

## 'GD0' (Unlock date and time setting)

Parameters: None

Format: 'GD0'

Output: '@'

Purpose: Unlock date and time setting. User can change date and time from KDC300 menu.

## 'GD1' (Lock date and time setting)

Parameters: None

Format: 'GD1'

Output: '@'

Purpose: Lock date and time setting. User can not change date and time from KDC300 menu.

## 'GM' (Message start position)

Parameters: row # and column #

Format: 'GM<row>;<column>#', where  $0 \leq \text{row} \leq 3$ ,  $0 \leq \text{column} \leq 12$  and # means the end of data

Row/Column	Character	Hex No.
0~9	"0" ~ "9"	30 ~ 39
10~12	"1", "0" ~ "1", "2"	31,30 ~ 31,32

Output: '!' – Invalid parameter

'@' – Set start position successfully

Purpose: To set start position of message

[Note] Default position is the 1<sup>st</sup> row and the 1<sup>st</sup> column. KDC maintains start position value until device is reset

[Example] Send 6 bytes (0x47 0x4D 0x31 0x3B 0x31 0x23) to set the message start position to the first row and the first column

- 0x47: ASCII code of character "G" in hexadecimal
- 0x4D: ASCII code of character "M" in hexadecimal
- 0x31: ASCII code of character "1" in hexadecimal
- 0x3B: ASCII code of character ":" in hexadecimal
- 0x31: ASCII code of character "1" in hexadecimal
- 0x23: ASCII code of character "#" in hexadecimal
- GM1;1#

## 'GMB0' (Failure Alert Beep)

Parameters: None

Format: 'GMB0'

Output: '@'

Purpose: To generates failure alert beep

## 'GMB1' (Success Alert Beep)

Parameters: None

Format: 'GMB1'

Output: '@'

Purpose: To generates success alert beep

## 'GMS' (Message display duration)

Parameters: Message display duration in seconds

Format: 'GMS<sec>#, where  $1 \leq \text{sec} \leq 60$  and # means the end of <sec>

Sec	Character	Hex No.
1~9	"1" ~ "9"	31 ~ 39
10~15	"A" ~ "F"	41 ~ 46
16~25	"1", "0" ~ "1", "9"	31,30 ~ 31,39
26~31	"1", "A" ~ "1", "F"	31,41 ~ 31,46
32~41	"2", "0" ~ "2", "9"	32,30 ~ 32,39
42~47	"2", "A" ~ "2", "F"	32,41 ~ 32,46
48~57	"3", "0" ~ "3", "9"	33,30 ~ 33,39
58~60	"3", "A" ~ "3", "C"	33,41 ~ 33,43

Output: '!' – Invalid parameter

'@' – Set start position successfully

Purpose: To set the message display duration

[Note] Default display duration is system sleep time. KDC maintains the display duration until device is reset

[Example] Send 6 bytes (0x47 0x4D 0x53 0x33 0x43 0x23) to set the message display duration to 60 seconds.

- 0x47: ASCII code of character "G" in hexadecimal
- 0x4D: ASCII code of character "M" in hexadecimal
- 0x53: ASCII code of character "S" in hexadecimal
- 0x33: ASCII code of character "3" in hexadecimal
- 0x43: ASCII code of character "C" in hexadecimal
- 0x23: ASCII code of character "#" in hexadecimal
- GMS3C#

## 'GMT' (Display message)

Parameters: Character string to be displayed on KDC

Format: 'GMT<message><CR>', where message is limited to 52 characters

Output: '@'

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.

[Example] Send 6 bytes (0x47 0x4D 0x55 0x61 0x62 0x0D) to display "ab"

- 0x47: ASCII code of character "G" in hexadecimal
- 0x4D: ASCII code of character "M" in hexadecimal
- 0x55: ASCII code of character "T" in hexadecimal
- 0x61: ASCII code of character "a" in hexadecimal
- 0x62: ASCII code of character "b" in hexadecimal
- 0x0D: ASCII code of <CR> in hexadecimal
- GMTab<CR>

## 'GSO' (Finish Synchronization)

Parameters: None

Format: 'GSO'

Output: '@'

Purpose: Acknowledge the finish of Synchronization process. KDC resumes normal mode upon receiving finish synchronization command.

## 'GS1' (Start Synchronization)

Parameters: None

Format: 'GS1'

Output: '@'

Purpose: Acknowledge the start of Synchronization process. KDC becomes synchronization mode, lock buttons and accept only "N", "W", "p", "G" commands from host.

## GTS' (Set termination character)

Parameters: 0, 1, 2, 3 or 4 in character or 0x30, 0x31, 0x32 or 0x33 in hexadecimal number

Format: 'GTS<value>#, where <value> is between 0 and 4, and # means the end of <value>

value	Termination Character	Character	Hex No.
0	None	"0"	30
1	CR	"1"	31
2	LF	"2"	32
3	CR/LF	"3"	33
4	Tab	"4"	34

Output: '!' – Invalid parameter

'@' – Set termination character successfully

Purpose: To set termination character after barcode data if data format is barcode only

[Example] Send 5 bytes (0x47, 0x54, 0x53, 0x30, 0x23) to set the termination character to NONE

- 0x47: ASCII code of character "G"
- 0x54: ASCII code of character "T"
- 0x53: ASCII code of character "S"
- 0x30: Set termination character is "NONE", ASCII code of character "0"
- 0x23: ASCII code of character "#"
- GTS0#

## 'GTG' (Read termination character)

Parameters: None

Format: 'GTG'

Output: '&lt;value&gt;@', where &lt;value&gt; is between 0 and 4, and @ means the end of &lt;value&gt;

value	Termination Character	Character	Hex No.
0	NONE	"0"	30
1	CR	"1"	31
2	LF	"2"	32
3	CR/LF	"3"	33
4	Tab	"4"	34

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

[Example] Output: 0x30, 0x40

- 0x30: ASCII code of character "0" in hexadecimal
- 0x40: ASCII code of character "@" in hexadecimal

## 'GW' (Menu Password Protection)

Parameters: 0 or 1<password>

Format: "GW0" or "GW1<password>#", where <password> is 5 U(p),D(down),S(can) key combination.

value	Note	Character	Hex No.
0	None	"0"	30
U	Up	"U"	55
D	Down	"D"	44
S	Scan	"S"	53

Output: '@'

Purpose: Enable or Disable menu password to prevent user access KDC300 menu

[Note] User should enter menu password in 5 seconds

[Example] Send 8 bytes data (0x47, 0x57, 0x55, 0x44, 0x55, 0x44, 0x53, 0x23) to enable handshake mode

- 0x47: ASCII code of character "G"
- 0x57: ASCII code of character "W"
- 0x55: ASCII code of character "U"
- 0x44: ASCII code of character "D"
- 0x55: ASCII code of character "U"
- 0x44: ASCII code of character "D"
- 0x53: ASCII code of character "S"
- 0x23: ASCII code of character "#"
- GWUDUDS#

## 'GX0' (Disable Auto Menu Exit)

Parameters: None

Format: 'GX0'

Output: '@'

Purpose: Disable automatic menu exit feature

## 'GX1' (Enable Auto Menu Exit)

Parameters: None

Format: 'GX1'

Output: '@'

Purpose: Enables KDC300 to exit menu mode. KDC300 automatically exits from menu mode if it remains idle in menu mode for 5 minutes.

## 'GY0' (Display data, type, date and time)

Parameters: None

Format: 'GY0'

Output: '@'

Purpose: Display barcode data, barcode type, date and time after scanning a barcode

## 'GY1' (Display data, date and time, battery status)

Parameters: None

Format: 'GY1'

Output: '@'

Purpose: Display barcode data, date and time, battery status after scanning a barcode

## 'GY2' (Display data, type, battery status)

Parameters: None

Format: 'GY2'

Output: '@'

Purpose: Display barcode data, barcode type, battery status after scanning a barcode

## 'H' (Set handshake mode)

Parameters: 0 or 1 in hexadecimal

Format: "H<value>#", where <value> is either 0 or 1 and # means the end of <value>

value	Handshake Mode	Character	Hex No.
0	Disable	"0"	30
1	Enable	"1"	31

Output: '!' – invalid parameter

'@' – all other times

Purpose: To Enable/Disable handshake with HOST while sending packet data. KD200P doesn't perform handshake if data format is barcode only.

### [Note]

- Handshake mode is applicable for only 'p' command and wedging in packet data format
- Handshake mode is NOT applicable for 'P' command and wedging in data only format
- If handshake mode is enabled,
  - KDC300 expects HOST to send either success or failure response to KDC300 once HOST receives packet data from KDC300.
  - 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 KDC300, HOST should send a failure response to KDC300 by sending a character '!'.
  - KDC300 tries multiple transmission up to 10 times if KDC300 doesn't get response within 1 second or receives failure response '!' from HOST.

[Example] Send 3 bytes data (0x48, 0x31, 0x23) to enable handshake mode

- 0x48: ASCII code of character "H"
- 0x31: ASCII code of character "1"
- 0x23: ASCII code of character "#"
- H1#

## 'h' (Read handshake mode)

Parameters: None

Format: 'h'

Output: '<value>@' where <value> is either 0 or 1 in hexadecimal, and @ means the end of <value>

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

[Example] Output: 0x00, 0x00, 0x00, 0x01, 0x40

- 0x0001: Handshake mode is 1
- 0x40: ASCII code of character "@" in hexadecimal

## 'L' (Set minimum barcode length)

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

Format: "L<value>#" where <value> is defined in the following table and # means the end of <value>

Minimum Length	Character	Hex. Number
2~9	"2" ~ "9"	32 ~ 39
10~15	"A" ~ "F"	41 ~ 46
16~25	"1", "0" ~ "1", "9"	31,30 ~ 31,39
26~31	"1", "A" ~ "1", "F"	31,41 ~ 31,46
32~36	"2", "0" ~ "2", "4"	32,30 ~ 32,34

Output: '!' – invalid parameter

'@' – all other times

Purpose: Set the minimum length of barcodes

[Example] Send 4 bytes data (0x4C, 0x31, 0x34, 0x23) to set the minimum barcode length to 20.

- 0x4C: ASCII code of character "L" in hexadecimal
- 0x31: ASCII code of character "1" in hexadecimal
- 0x34: ASCII code of character "4" in hexadecimal
- 0x23: ASCII code of character "#" in hexadecimal
- L14#

## 'I' (Read minimum barcode length)

Parameters: None

Format: 'I'

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

Purpose: To get the current setting of minimum barcode length

[Example] Output: 0x00, 0x00, 0x00, 0x0B, 0x40

- 0x000B is the minimum barcode length (11 in decimal)
- 0x40 is the ASCII code of character "@" in hexadecimal

## 'M' (Read serial number)

Parameters: None

Format: 'M'

Output: 'yymmxxxxx@' where yy is manufacture year and month, xxxxxx is the serial number and @ means the end of serial number

value	Character	Hex No.
0	"0"	30
1	"1"	31
2	"2"	32
3	"3"	33
4	"4"	34
5	"5"	35
6	"6"	36
7	"7"	37
8	"8"	38
9	"9"	39

Purpose: To get the serial number of KDC300

[Example] Output: 0x30, 0x38, 0x30, 0x33, 0x30, 0x30, 0x31, 0x32, 0x37, 0x39, 0x40

- 0x30: ASCII code of character "0" in hexadecimal
- 0x38: ASCII code of character "8" in hexadecimal
- 0x30: ASCII code of character "0" in hexadecimal
- 0x33: ASCII code of character "3" in hexadecimal
- 0x30: ASCII code of character "0" in hexadecimal
- 0x30: ASCII code of character "0" in hexadecimal
- 0x31: ASCII code of character "1" in hexadecimal
- 0x32: ASCII code of character "2" in hexadecimal
- 0x37: ASCII code of character "7" in hexadecimal
- 0x39: ASCII code of character "9" in hexadecimal
- 0x40: ASCII code of character "@" in hexadecimal
- 0803001279@

## 'N' (Get # of stored barcode)

Parameters: None

Format: 'N'

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

Purpose: To get the number of stored barcodes in KDC300 internal flash memory

[Example] Output: 0x00, 0x00, 0x01, 0x1C, 0x40

- Ignore the first two leading NULL bytes
- 0x11C (284 in decimal) stored barcodes
- 0x40 is the ASCII code of character "@" in hexadecimal

## 'O' (Set barcode option)

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

Format: 'O<1<sup>st</sup> option value>#<2<sup>nd</sup> option value>#' where <value> is the value of parameter in hexadecimal and # means the end of <value>

Output: '!' – invalid parameter

'@' – all other times

Purpose: To enable the selected options

[Example1] Send 5 bytes data (0x4F, 0x30, 0x23, 0x30, 0x23) to disable all options

- 0x4F: ASCII code of character "O" in hexadecimal
- 0x30: ASCII code of character "0" in hexadecimal
- 0x23: ASCII code of character "#" in hexadecimal
- 00#0#

[Example2] Send 19 bytes (0x4F, 0x46, 0x46, 0x46, 0x46, 0x46, 0x46, 0x46, 0x46, 0x46, 0x23, 0x46, 0x46, 0x46, 0x46, 0x46, 0x46, 0x46, 0x23) to enable all options

- 0x4F: ASCII code of character "O" in hexadecimal
- 0x46: ASCII code of character "F" in hexadecimal
- 0x23: ASCII code of character "#" in hexadecimal
- OFFFFFFFF#FFFFFFFF#

[Note] The following table shows how each bit is assigned on 1<sup>st</sup> and 2<sup>nd</sup> option values.

1<sup>st</sup> option values

Barcodes	bit#	Options
<b>Codabar</b>	0	0: Do not transmit start/stop char. 1: Transmit start/stop char
	1	00: Do not verify check char.
	2	01: Verify check digit but do not transmit 10: Verify check digit and transmit
	3	00: Concatenation off
	4	01: Concatenation on 10: Concatenation require
<b>Reserved</b>	5	Not used
	6	Not used
	7	Not used

<b>Code 39</b>	8	0: Do not transmit start/stop char. 1: Transmit start/stop char
	9	00: Do not verify check char.
	10	01: Verify check digit but do not transmit 10: Verify check digit and transmit
	11	0: Append off 1: Append on
	12	0: Full ASCII off 1: Full ASCII on
<b>Reserved</b>	13	Not used
	14	Not used
	15	Not used
<b>I2of5</b>	16	00: Do not verify check char.
	17	01: Verify check digit but do not transmit 10: Verify check digit and transmit
<b>Reserved</b>	18	Not used
	19	Not used
<b>Code 11</b>	20	0: Verify check digit 1: Verify check digits
<b>Code128</b>	21	0: ISBT concatenation off 1: ISBT concatenation on
<b>Telepen</b>	22	0: AIM output 1: Original output
<b>Reserved</b>	23	Not used
	24	Not used
	25	Not used
	26	Not used
	27	Not used
	28	Not used
	29	Not used
<b>PosiCode</b>	30	00: A and B On
	31	01: A and B and Limited A On 10: A and B and Limited B On

## 2nd option values

Barcodes	bit#	Options
<b>UPCA</b>	0	0: Do not verify check digit 1: Verify check digit
	1	0: Number system off 1: Number system on
	2	0: 2 digit addenda off 1: 2 digit addenda on
	3	0: 5 digit addenda off 1: 5 digit addenda on
	4	0: Addenda not required 1: Addenda required
	5	0: Addenda separator off 1: Addenda separator on
	6	0: Extended coupon code off 1: Extended coupon code on
<b>UPCE</b>	7	0: Expand off 1: Expand on
	8	0: Addenda not required 1: Addenda required
	9	0: Addenda separator off 1: Addenda separator on
	10	0: Check digit off 1: Check digit on
	11	0: Number system off 1: Number system on
	12	0: 2 digit addenda off 1: 2 digit addenda on
	13	0: 5 digit addenda off 1: 5 digit addenda on
<b>EAN-13</b>	14	0: Do not verify check digit 1: Verify check digit
	15	0: 2 digit addenda off 1: 2 digit addenda on

	16	0: 5 digit addenda off 1: 5 digit addenda on
	17	0: Addenda not required 1: Addenda required
	18	0: Addenda separator off 1: Addenda separator on
	19	0: ISBN translate off 1: ISBN translator on
<b>Postnet</b>	20	0: Check digit do not transmit 1: Check digit transmit
<b>PlanetCode</b>	21	0: Check digit do not transmit 1: Check digit transmit
<b>EAN-8</b>	22	0: Do not verify check digit 1: Verify check digit
	23	0: 2 digit addenda off 1: 2 digit addenda on
	24	0: 5 digit addenda off 1: 5 digit addenda on
	25	0: Addenda not required 1: Addenda required
	26	0: Addenda separator off 1: Addenda separator on
<b>MSI</b>	27	0: Verify check character, but do not transmit 1: Verify check character and transmit
<b>EANUCC</b>	28	0: Version off 1: Version on
	29	00: Emulation off
	30	01: 128 emulation 10: RSS emulation
<b>Reserved</b>	31	Not used

## 'o' (Read barcode option)

Parameters: None

Format: 'o'

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

Purpose: To get the barcode options saved in KDC300. Please see the table of command 'O' for the detailed explanation of options

[Example] Output: 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x40

- 0xFFFFFFFFFFFFFFFF: All options are enabled
- 0x40: ASCII code of character "@" in hexadecimal

## 'P' (Download all stored barcode)

Parameters: None.

Format: 'P'

Output: A multi-byte string (see below)

Purpose: Get all barcodes data stored in KDC300 internal flash memory. 'P' command does **not** erase the stored barcodes.

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

N	C <sub>0</sub>	Y <sub>0</sub>	D <sub>0</sub>	T <sub>0</sub>	C <sub>1</sub>	Y <sub>1</sub>	D <sub>1</sub>	T <sub>1</sub>	.	.	.	.	.	.	.	.	.	.	.	.	C <sub>n</sub>	Y <sub>n</sub>	D <sub>n</sub>	T <sub>n</sub>
---	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	---	---	---	---	---	---	---	---	---	---	---	---	----------------	----------------	----------------	----------------

Where

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

C<sub>0</sub>, C<sub>1</sub>,..., C<sub>n</sub> : Total number of each barcode data(2 bytes) including C/Y/D/T bytes

Y<sub>0</sub>,Y<sub>1</sub>,..., Y<sub>n</sub> : Type of each barcode(1 byte)

- Bit 0 ~5 : Barcode type

Symbology	Value	Symbology	Value
Code 32	0	I2of5	22
Trioptic	1	IATA	23
Korea Post	2	MSI	24
Aus. Post	3	Code 11	25
British Post	4	Code 93	26
Canada Post	5	Code 128	27
EAN-8	6	Code 49	28
UPC-E	7	Matrix2of5	29
UCC/EAN-128	8	Plessey	30
Japan Post	9	Code 16K	31
KIX Post	10	CodablockF	32
Planet Code	11	PDF417	33
OCR	12	QR/Micro QR	34
Postnet	13	Telepen	35
China Post	14	VeriCode	36
Micro PDF417	15	Data Matrix	37
TLC 39	16	MaxiCode	38

PosiCode	17	EAN/UCC	39
Codabar	18	RSS	40
Code 39	19	Aztec Code	41
UPC-A	20	No Read	42
EAN-13	21	Unknown	43

– Bit 6/7

00: Normal barcode

01: Mismatched slave barcode

10: Matched slave barcode

11: Master barcode

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

T0, T1,...,Tn : 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 3<sup>rd</sup> 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] Output: 0x00, 0x00, 0x16, 0x00, 0x12, 0x14, 0x30, 0x31, 0x32, 0x35, 0x34, 0x36, 0x36, 0x31, 0x32, 0x32, 0x39, 0x36, 0x00, 0x42, 0x17, 0x24

- N: 0x00, 0x00, 0x16 : 22 bytes
- C0: 0x12 : 18 bytes
- Y0: 0x14 : UPCA
- D0: 0x30, 0x31, 0x32, 0x35, 0x34, 0x36, 0x36, 0x31, 0x32, 0x32, 0x39, 0x36 : 012546612296 UPCA barcode
- T0: 0x00, 0x42, 0x17, 0x24 :

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' (Download n<sup>th</sup> stored barcode)

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 and # means the end of <value>. The first stored barcode location is "0", not "1"

Barcode Position	Character	Hex No.
0~9	"0" ~ "9"	30 ~ 39
10~15	"A" ~ "F"	41 ~ 46
16~25	"1", "0" ~ "1", "9"	31,30 ~ 31,39
26~31	"1", "A" ~ "1", "F"	31,41 ~ 31,46
...	...	...
10240	"2", "8", "0", "0"	32,38,30,30

Output: A multi-byte string defined in the following note.

Purpose: Get the Nth barcode data stored in KDC300 memory. 'p' command does not erase the barcode in memory.

[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)

– Bit 0 ~45: Barcode type

Symbology	Value	Symbology	Value
Code 32	0	I2of5	22
Trioptic	1	IATA	23
Korea Post	2	MSI	24
Aus. Post	3	Code 11	25
British Post	4	Code 93	26
Canada Post	5	Code 128	27
EAN-8	6	Code 49	28
UPC-E	7	Matrix2of5	29
UCC/EAN-128	8	Plessey	30
Japan Post	9	Code 16K	31

KIX Post	10	CodablockF	32
Planet Code	11	PDF417	33
OCR	12	QR/Micro QR	34
Postnet	13	Telepen	35
China Post	14	VeriCode	36
Micro PDF417	15	Data Matrix	37
TLC 39	16	MaxiCode	38
PosiCode	17	EAN/UCC	39
Codabar	18	RSS	40
Code 39	19	Aztec Code	41
UPC-A	20	No Read	42
EAN-13	21	Unknown	43

– Bit 6/7

00: Normal barcode

01: Mismatched slave barcode

10: Matched slave barcode

11: Master barcode

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)
- KDC300 tries multiple transmission up to 10 times if Handshake mode is enabled and KDC300 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] Send 4 bytes data (0x70, 0x31, 0x34, 0x23) to KDC to upload the barcode data stored in 20<sup>th</sup> position

- 0x70: ASCII code of character “p” in hexadecimal
- 0x31: ASCII code of character “1” in hexadecimal
- 0x34: ASCII code of character “4” in hexadecimal

- 0x23: ASCII code of character “#” in hexadecimal
- P14#

## 'S' (Select symbology)

Parameters: The symbology selection

Format: 'S<value>#<value>#', where the length of <value> is between 1 and 8, and # means the end of <value>

- The length of <value> is between 1 to 8
  - Compute <value> by 4 bits, up to 32 bits
  - Send the most significant 4 bits first. The least significant 4 bits will be sent before “#” character (0x23 in hexadecimal)

value	Character	Hex No.
0(0x0) ~ 9(0x9)	"0" ~ "9"	30 ~ 39
10(0xA) ~ 15(0xF)	"A" ~ "F"	41 ~ 46

- The <value> enable/disable symbology
- Set the appropriate bit to “1” to enable symbology

### 1D Symbologies

Group	bit#	Symbology
1D Symbologies	0	Codabar
	1	Code 32
	2	Code 39
	3	Code 93
	4	Code 11
	5	Code 128
	6	EAN-13
	7	EAN-8
	8	EAN/UCC
	9	Interleaved 2 of 5
	10	Matrix 2 of 5
	11	MSI
	12	Plessy code
	13	PosiCode
	14	RSS-14
15	RSS Limited	

	16	RSS Expanded
	17	Straight 2 of 5 Industrial
	18	Straight 2 of 5 IATA
	19	TCIF Linked Code 39(TLC39)
	20	Telepen
	21	Trioptic
	22	UPC-A
	23	UPC-E0
	24	UPC-E1
<b>Reserved</b>	25	Not used
	26	Not used
	27	Not used
	28	Not used
	29	Not used
	30	Not used
	31	Not used

## 2D, Postal and OCR

Group	bit#	Symbology
<b>2D symbologies</b>	0	Aztec code
	1	Aztec Runes
	2	Codablock F
	3	Code 16K
	4	Code 49
	5	Data Matrix
	6	MaxiCode
	7	MicroPDF417
	8	PDF417
	9	QR Code
<b>Reserved</b>	10	Not used
	11	Not used
<b>Postal codes</b>	12	Postnet

	13	Planet Code
	14	British Post
	15	Canadian Post
	16	Kix(Netherlands) Post
	17	Australian Post
	18	Japanese Post
	19	China Post
	20	Korea Post
<b>Reserved</b>	21	Not used
	22	Not used
	23	Not used
<b>OCR</b>	24	OCR Off
	25	OCR-A
	26	OCR-B
	27	OCR U.S. Currency
	28	OCR MICR E-13B
	29	OCR Semi Font
<b>Reserved</b>	30	Not used
	31	Not used

Output: '!' – invalid parameter

'@' – all other times

Purpose: To enable/Disable symbologies

[Example1] Send 5 bytes data (0x53, 0x30, 0x23, 0x30, 0x23) to disable all the options

- 0x53: ASCII code of character "S" in hexadecimal
- 0x30: ASCII code of character "0" in hexadecimal
- 0x23: ASCII code of character "#" in hexadecimal
- S0#0#

[Example2] Send 19 bytes (0x53, 0x46, 0x46, 0x46, 0x46, 0x46, 0x46, 0x46, 0x46, 0x46, 0x23, 0x46, 0x46, 0x46, 0x46, 0x46, 0x46, 0x46, 0x23) to enable all options

- 0x53: ASCII code of character "O" in hexadecimal
- 0x46: ASCII code of character "F" in hexadecimal
- 0x23: ASCII code of character "#" in hexadecimal
- SFFFFFFFF#FFFFFFFF#

## 's' (Read selected symbology)

Parameters: None

Format: 's'

Output: '<value>@', where <value> is 8 bytes and @ means the end of <value>;

*<value> should be interpreted according to the table described in "S" command.*

Purpose: To get enabled/disabled symbology options

[Example] Output: 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x40

- 0xFFFFFFFFFFFFFFFF: All options are enabled
- 0x40: ASCII code of character "@" in hexadecimal

## 'T' (Set Timeout)

Parameters: Time in msec.

Format: "T<value>#", where <value> is 1 to 4 bytes and # means the end of <value>

- <value> is between 100msec to 10sec (10000msec)

Timeout	Character	Hex No.
500 (0x1F4) msec	"1", "F", "4"	31,46,34
1sec(0x3E8 msec)	"3", "E", "8"	33,45,38
...	...	...
10sec(0x2710 msec)	"2", "7", "1", "0"	32,37,31,30

Output: '!' – invalid parameter

'@' – all other times

Purpose: To set the timeout for barcode read operation

[Example] Send 6 bytes data (0x54, 0x32, 0x37, 0x,31, 0x30, 0x23) to set timeout to 10sec.

- 0x54: ASCII code of character "T" in hexadecimal
- 0x32: ASCII code of character "2" in hexadecimal
- 0x37: ASCII code of character "7" in hexadecimal
- 0x31: ASCII code of character "1" in hexadecimal
- 0x30: ASCII code of character "0" in hexadecimal
- 0x23: ASCII code of character "#" in hexadecimal
- T2710#

## 't' (Read timeout value)

Parameters: None

Format: 't'

Output: '<value>@' where <value> is the timeout value for barcode read operation in hexadecimal  
@ means the end of <value>

Purpose: To get the current read timeout setting

[Example] Output: 0x00, 0x00, 0x07, 0xD0, 0x40

- 0x007D0: 10000 in decimal, 10sec (10000msec)
- 0x40: ASCII code of character "@" in hexadecimal

## 'U' (Set data process mode)

Parameters: 0, 1, 2, 3 or 4

Format: "U<value>#", where <value> is between 0 and 4, and # means the end of <value>

value	Character	Hex No.
0	"0"	30
1	"1"	31
2	"2"	32
3	"3"	33
4	"4"	34

- 0: KDC300 doesn't store scanned barcode in the memory and just transmits it to the host
- 1: KDC300 stores scanned data in the memory and transmits it to the host
- 2: KDC300 stores scanned data in the memory but doesn't transmit it to the host
- 3: KDC300 stores scanned data in the memory only if it succeeds to send read barcode to the host
- 4: KDC300 stores scanned data in the memory only if it fails to send read barcode to the host

Output: '!' – invalid parameter

'@' – all other times

Purpose: To set the data process mode of KDC300

## 'u' (Read data process mode)

Parameters: None

Format: 'u'

Output: '<value>@' where <value> is barcode read operation mode value in hexadecimal, and @ means the end of <value>

- 0: KDC300 doesn't store scanned barcode in the memory and just transmits it to the host
- 1: KDC300 stores scanned data in the memory and transmits it to the host
- 2: KDC300 stores scanned data in the memory but doesn't transmit it to the host
- 3: KDC300 stores scanned data in the memory only if it succeeds to send read barcode to the host
- 4: KDC300 stores scanned data in the memory only if it fails to send read barcode to the host

Purpose: Get the current setting of barcode handling mode

[Example] Output: 0x00, 0x00, 0x00, 0x01, 0x40

- 0x0001: barcode read operation mode is 1
- 0x40: ASCII code of character "@" in hexadecimal

## 'V' (Read firmware version)

Parameters: None

Format: "V"

Output: '<string>@' where <string> is KDC300 firmware version and @ means the end of data

Purpose: To read the firmware version of KDC300

## 'W' (Wake up)

Parameters: None

Format: "W"

Output: '@'

Purpose: To wake up KDC300 from sleep mode

[Note] KDC300 will not respond to "W" command if it is in sleep mode. It is required to send 'W' command multiple times until receiving '@' output.

## 'w' (Set data format)

Parameters: 0 or 1

Format: 'w<value>#', where <value> is either 0 or 1, and # means the end of <value>

value	Character	Hex No.
0	"0"	30
1	"1"	31

- 0: KDC300 sends barcode only to HOST,
- 1: KDC300 sends PACKET data and wait for response('@') from HOST

Output: '!' – invalid parameter

'@' – all other times

Purpose: KDC300 transmits scanned data to HOST if Wedge mode is enabled. User can set Wedge mode using 'U' command. KDC300 can send barcode data only or PACKET data

### [Note]

- KDC300 tries multiple transmission up to 10 times if Handshake mode is enabled and KDC300 doesn't get response from HOST within 1 second
- KDC300 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' (Set security level)

Parameters: Security level

Format: "Z&lt;value&gt;#", &lt;value&gt; is between 1 and 4, and # means the end of &lt;value&gt;

value	Character	Hex No.
1	"1"	31
2	"2"	32
3	"3"	33
4	"4"	34

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

Output: '!' – invalid parameter

'@' – all other times

Purpose: To set the security level

[Example] Send 3 bytes data (0x5A, 0x31, 0x23) to set the security level to 1

- 0x5A: ASCII code of character "Z" in hexadecimal
- 0x31: ASCII code of character "1" in hexadecimal
- 0x23: ASCII code of character "#" in hexadecimal
- Z1#

## 'z' (Read security level)

Parameters: None

Format: 'z'

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

Purpose: Get the current security level value

[Example] Output: 0x00, 0x00, 0x00, 0x01, 0x40

- 0x0001: security level is 1
- 0x40: ASCII code of character "@" in hexadecimal

## Wedging Data Format and Handshake Mode

---

KDC300 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

- KDC300 does not perform handshaking if Wedging data format is packet data but handshake mode is disabled.

### Packet data format and enabled Handshake mode

- KDC300 expects the HOST to send either success or failure response to KDC300 once HOST receives packet data from KDC300 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 character '@'. If HOST detected an error while receiving a packet data from KDC300, HOST should send a failure response to KDC300 by sending a character '!'.
- KDC300 tries multiple transmission up to 10 times if Handshake mode is enabled and KDC300 doesn't get response from HOST within 1 second

## Miscellaneous

---

### KDC300 sends "?" character

"?" is sent by KDC300 when the device receives undefined character (command) from Host.  
Application ignores "?" character

### KDC300 sends one to three leading "NULL" bytes

KDC300 *Bluetooth* module goes to sleep mode if not used for a while. KDC300 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

---



CORPORATE HEADQUARTERS  
116 Village Boulevard, Suite 200  
Princeton, NJ 08540  
PH: 609-734-4335 FAX: 609-228-4373  
email: [admin@koamtac.com](mailto:admin@koamtac.com)

**For more information, visit our website - [www.koamtac.com](http://www.koamtac.com)**