

ΚΟΛΜΤΑΣ 

KDC400
Programming Manual Draft
(Rev. 3.03/2.85.P/2.86.P)

November, 2011

Copyright, License and Warning page

Copyright® 2011 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.

May be covered under one or more pending and/or US Patents, including US Pat. No. 7,769,917 B1
 KDC® is a registered trademark and property of KoamTac, Inc.
 KoamTac® is a registered trademark and property of KoamTac, Inc.
 KTSync® is a registered trademark and property of KoamTac, Inc.

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.

WARNING: RISK OF EXPLOSION IF BATTERY IS REPLACED BY AN INCORRECT TYPE. DISPOSE USED BATTERIES ACCORDING TO THE INSTRUCTIONS.

Table of Contents

Internal data buffer	7
<i>Data Format</i>	7
<i>How many barcodes can be stored?</i>	9
<i>What happens when the buffer is full?</i>	9
Lock & Unlock KDC Commands	10
<i>KDC Commands Lock</i>	10
<i>KDC Commands Unlock</i>	10
AES Key.....	11
Secondary <i>Bluetooth</i> Module.....	12
<i>Connect/Disconnect</i>	12
<i>Send/Receive Data to/from Printer</i>	13
<i>Test Application</i>	13
Commands Set	14
' <i>B</i> ' (<i>Read battery level</i>).....	22
' <i>bSC</i> ' (<i>Connect to printer</i>).....	23
' <i>bSD</i> ' (<i>Send data to printer</i>).....	24
' <i>bSN</i> ' (<i>Discover printers and send list</i>).....	25
' <i>bSS</i> ' (<i>Get printer connection status</i>).....	26
' <i>bT1</i> ' (<i>Bluetooth power on/off</i>)	28
' <i>bT2</i> ' (<i>Bluetooth pairing Mode</i>).....	29
' <i>bT3</i> ' (<i>Bluetooth auto connection</i>).....	30
' <i>bT4</i> ' (<i>Bluetooth auto power on</i>).....	31
' <i>bT5</i> ' (<i>Bluetooth auto power off</i>).....	32
' <i>bT6</i> ' (<i>Bluetooth disconnection beep warning</i>).....	33
' <i>bT70</i> ' (<i>Read Bluetooth auto power off timeout value</i>).....	34
' <i>bT71</i> ' (<i>Set Bluetooth auto power off timeout value</i>)	35
' <i>bT80</i> ' (<i>Disable Bluetooth power off message to host</i>)	36
' <i>bT81</i> ' (<i>Enable Bluetooth power off message to host</i>).....	37
' <i>bT9</i> ' (<i>Get Bluetooth MAC address</i>).....	38
' <i>bTA0</i> ' (<i>Set KDC400 in Bluetooth Slave Mode</i>).....	39
' <i>bTA1</i> ' (<i>Set KDC400 in Bluetooth Master Mode</i>).....	40
' <i>bTc0</i> ' (<i>Set Bluetooth ConnectDevice to SPP</i>).....	41
' <i>bTc1</i> ' (<i>Set Bluetooth ConnectDevice to HID iOS</i>)	42
' <i>bTc2</i> ' (<i>Set Bluetooth ConnectDevice to iPhone</i>)	43
' <i>bTc3</i> ' (<i>Set Bluetooth ConnectDevice to SPP2.0</i>).....	44
' <i>bTc4</i> ' (<i>Set ConnectDevice to HID normal</i>).....	45
' <i>bTcG</i> ' (<i>Return ConnectDevice mode</i>)	46
' <i>bTC</i> ' (<i>Activates a connection to predefined Bluetooth MAC address and PIN code</i>).....	47

'bTD' (Set Bluetooth MAC address).....	48
'bTE' (Set Bluetooth PIN code).....	49
'bTO0' (Read Bluetooth auto power on delay time).....	50
'bTO1' (Set Bluetooth auto power on delay time).....	51
'bTS0' (Disable ConfirmToSend).....	52
'bTS1' (Enable ConfirmToSend).....	53
'bTV' (Bluetooth Firmware Version).....	54
'bTW0' (Disable Leading Null bytes).....	55
'bTW1' (Enable Leading Null bytes).....	56
'C' (Set date and time).....	57
'c' (Read date and time).....	58
'D' (Software trigger).....	59
'E' (Erase memory).....	60
'e' (Erase the last stored data).....	61
'F' (Factory default).....	62
'GB0' (Unlock button).....	63
'GB1' (Lock button).....	64
'Gb0' (Beeper off).....	65
'Gb1' (Beeper on).....	66
'Gb2' (Get Beep sound setting).....	67
'Gb3' (Get Beep volume setting).....	68
'GbV0' (Low beep volume).....	69
'GbV1' (High beep volume).....	70
'GC0' (Disable port connection status display).....	71
'GC1' (Enable port connection status display).....	72
'GD0' (Unlock date and time setting).....	73
'GD1' (Lock date and time setting).....	74
'GESP' (Set Prefix String).....	75
'GESS' (Set Suffix String).....	76
'GESA' (Add or Remove AIM ID).....	77
'GEGP' (Get Prefix String).....	78
'GEGS' (Get Suffix String).....	79
'GEGA' (Get AIM ID Setting).....	80
'GESO' (Set Start Position of Partial Data).....	81
'GEGO' (Get Start Position of Partial Data).....	82
'GESL' (Set Length of Partial Data).....	83
'G EGL' (Get Length of Partial Data).....	84
'GEST' (Set Partial Data Action Flag).....	85
'GEGT' (Get Partial Data Action Flag).....	86
'GHKG' (Get HID Keyboard layout).....	87

'GHKS' (Set HID Keyboard layout).....	88
'GHTG' (Get HID Autolock Time).....	89
'GHTS' (Set HID Autolock Time).....	90
'GIDG' (Get current pixel depth).....	91
'GIDS' (Set current pixel depth of BMP format).....	92
'GIFG' (Get current image format).....	93
'GIFS' (Set current image format).....	94
'GIMSO' (Disable image capture).....	95
'GIMS1' (Enable image capture).....	96
'GIMG' (Get image capture setting).....	97
'GINH' (Snap and ship image).....	98
'GKG' (Get current KDC Mode).....	99
'GKS' (Set current KDC Mode).....	100
'GL0' (Disable Display Scroll).....	101
'GL1' (Enable Display Scroll).....	102
'GM' (Message start position).....	103
'GMB0' (Failure Alert Beep).....	104
'GMB1' (Success Alert Beep).....	105
'GMBC' (Custom beep tone).....	106
'GMC0' (Clear screen).....	107
'GMd' (Display bitmap without clear screen).....	108
'GMD' (Display bitmap after clear screen).....	109
'GMf' (Set message font size).....	110
'GMS' (Message display duration).....	111
'GMT' (Display message).....	112
'GnCG' (Get HID control character mode).....	113
'GnCS' (Set HID Control Character).....	114
'GndBG' (Get HID initial delay).....	115
'GndBS' (Set HID initial delay).....	116
'GndCG' (Get HID inter-character delay).....	117
'GndCS' (Set HID inter-character delay).....	118
'GndG' (Get duplicate check setting in normal mode).....	119
'GnDS0' (Disable duplicate check in normal mode).....	120
'GnDS1' (Enable duplicate check in normal mode).....	121
'GnEG' (Get current auto erase setting of Auto Erase).....	122
'GnES0' (Disable auto erase).....	123
'GnES1' (Enable auto erase).....	124
'GnF' (Get current Data Format).....	125
'GnMBG' (Get MSR error beep tone).....	126
'GnMBS' (Set MSR error beep tone).....	127

'GnMDG' (Get MSR Data Type).....	128
'GnMDS' (Set MSR Data Type).....	129
'GnMEG' (Get MSR Data Encrypt Mode).....	130
'GnMES' (Set MSR data encrypt mode).....	131
'GnMES' (Set MSR data encrypt mode).....	132
'GnMkG' (Get AES Key length).....	133
'GnMkS' (Set AES Key length).....	134
'GnMKG' (Get AES Key).....	135
'GnMKS' (Set AES Key).....	136
'GnMSS' (Set MSR track separator).....	137
'GnMTG' (Get MSR track selection status).....	138
'GnMTS' (Set MSR track selection).....	139
'GnS1' (Get available flash memory size).....	140
'GnTG' (Get sleep timeout).....	141
'GnTS' (Set sleep timeout).....	142
'Gpm0' (Unlock Partial data display menu entry).....	143
'Gpm1' (Lock Partial data display menu entry).....	144
'Gps' (Partial data display start position).....	145
'Gpl' (Partial data display length).....	146
'Gpt' (Set Partial Display Action Flag).....	147
'GS0' (Finish Synchronization).....	148
'GS1' (Start Synchronization).....	149
'GTS' (Set termination character).....	150
'GTG' (Read termination character).....	151
'GtGD' (Get Auto Trigger Reread Delay).....	152
'GtGM' (Get Auto Trigger Mode).....	153
'GtSD' (Set Auto Trigger Reread Delay).....	154
'GtSM0' (Disable Auto Trigger Mode).....	155
'GtSM1' (Enable Auto Trigger Mode).....	156
'GW' (Menu Password Protection).....	157
'GX0' (Disable Auto Menu Exit).....	158
'GX1' (Enable Auto Menu Exit).....	159
'GY0' (Display data, type, date and time).....	160
'GY1' (Display data, date and time, battery status).....	161
'GY2' (Display data, type, battery status).....	162
'GY3' (Display memory status).....	163
'GY5' (Display Barcode only).....	164
'H' (Set handshake mode).....	165
'h' (Read handshake mode).....	166
'L' (Set minimum barcode length).....	167

'l' (Read minimum barcode length).....	168
'M' (Read serial number).....	169
'N' (Get # of stored barcode).....	170
'O' (Set barcode option) - KDC410/415 Model.....	171
'O' (Set barcode option) - KDC420/425 Model.....	172
'o' (Read barcode option) - KDC410/415 Model.....	176
'o' (Read barcode option) - KDC420/425 Model.....	177
'P' (Download all stored barcode) - KDC410/415 Model.....	178
'P' (Download all stored barcode) - KDC420/425 Model.....	180
'p' (Download n th stored barcode) - KDC410/415 Model.....	182
'p' (Download n th stored barcode) - KDC420/425 Model.....	184
'S' (Select symbology) - KDC410/415 Model.....	186
'S' (Select symbology) - KDC420/425 Model.....	188
's' (Read selected symbology) - KDC410/415 Model.....	191
's' (Read selected symbology) - KDC420/425 Model.....	192
'T' (Set Timeout).....	193
't' (Read timeout value).....	194
'U' (Set data process mode).....	195
'u' (Read data process mode).....	196
'V' (Read firmware version).....	197
'W' (Wake up).....	198
'w' (Set data format).....	199
'Z' (Set security level).....	200
'z' (Read security level).....	201
Wedging Data Format and Handshake Mode	202
Barcode only format.....	202
Packet data format and disabled Handshake mode.....	202
Packet data format and enabled Handshake mode.....	202
Miscellaneous	203
KDC400 sends "?" character.....	203
KDC400 sends one to three leading "NULL" bytes.....	203
Contact Information	204

KoamTac KDC400 provides easy and extensive commands set to application developers who wish to control the KDC400 and resulting data for their application.

Internal data buffer

KDC400 has 175K 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 175K 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

KDC400 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 record (Y+D+T) (1 byte).

2 bytes for KDC400, 1 byte for KDC100/200/250

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

- The 1st 3 bits distinguish Normal mode or Master/Slave Application records
 - Bit 5 : 1 if matching slave data, 0 if mismatching slave data
 - Bit 6 : 1 if master data, 0 if slave data
 - Bit 7 : 1 if data is collected in Master/Slave Application, 0 if data is collected in Normal mode
- The remaining 5 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 record (Y+D+T) (1 byte), then the barcode type (1 byte), then the barcode value (without the string terminator '\0'), and then timestamp(4 bytes).

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

KDC400 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 record (Y+D+T+Q) (1 byte).

2 bytes for KDC400, 1 byte for KDC100/200250

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

- o The 1st 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 1st byte indicates Predefined or User Application
 - 0xFF : User Application
- o The 2nd 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 record (Y+D+T+Q) (1 byte), then the barcode type (1 byte), then the barcode value (without the string terminator '\0'), then timestamp(4 bytes), and then quantity value(2 bytes)

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, KDC400 can store 10,240 UPCA barcodes.

What happens when the buffer is full?

When the internal flash memory is full, KDC400 displays “Buffer Full” message and ignores the command to scan barcodes.

Lock & Unlock KDC Commands

User can disable KDC commands to prevent accidental command string from host. We **strongly** recommend unlocking and locking KDC commands before sending individual KDC commands.

KDC supports only “W (Wakeup)” and Unlock commands if KDC commands are locked. User can upgrade KDC firmware regardless of KDC lock status.

KDC Commands Lock

Parameters: None

Format: <0x0F><K><0x03>

Purpose: Lock KDC commands

KDC Commands Unlock

Parameters: None

Format: <0x0F><K><0x02>

Purpose: Unlock KDC commands

AES Key

KDC415/425/430 support 128 bits, 192 bits and 256 bits AES key length. KDC provides a way to read current AES key length and to set a new AES key length.

KDC415/425/430 use AES Key to encrypt data. The AES key can be changed through KDC command. The following table shows AES key length and the number of AES key characters. Because KDC415/425/430 support up to 256 bits AES key length, the maximum number of AES key for KDC is 32 characters.

AES Key Length	Number of AES Key Character
128 bits	16
192 bits	24
256 bits	32

Secondary *Bluetooth* Module

KDC410i/420i has secondary Bluetooth module for printer and this document describes firmware and iOS SDK changes to support this secondary Bluetooth module. The name of secondary Bluetooth module is the same as primary Bluetooth module except 'S' is attached to the end of serial number as depicted 'KDC420[010020S]'.

Connect/Disconnect

There are two ways to connect *Bluetooth* printer via KDC's secondary *Bluetooth* module. One way is to send a command to KDC. The second way is to scan a special barcode. Before sending a command to KDC, it is also possible to discover available printers.

Discovering printers

Host application can ask KDC to discover available printers and to send the list to host. Host application displays the list and connects selected printer.

Sending a connect command

Host application can initiate printer connection by sending a command to KDC which includes printer MAC address. KDC tries to connect previous connected printer if no MAC address is provided.

Special barcode

KDC firmware provides a special barcode for connecting to a printer with MAC address or connecting to a previously connected printer.

Disconnect

Disconnection performed when powering off printer or KDC is out of the printer's range. It also can be done by powering off KDC's Bluetooth modules which will disconnect both host and printer connection.

Printer connection status

Host application requests KDC to reply whether printer is connected or not.

Send/Receive Data to/from Printer

Send data to printer

The main purpose of secondary *Bluetooth* module is to send data to a printer. KDC keeps connection with Bluetooth printer through secondary *Bluetooth* module while it is connected to HOST via primary Bluetooth module. When Host needs to print data, it sends data to KDC and KDC transfers this data to printer via the secondary *Bluetooth* module.

Once KDC has finished transferring host data to printer, it returns to normal mode.

Send printer response to host

KDC also transfers data comes from printer to host.

Test Application

iOS SDK provides following menus to demonstrate Bluetooth connection to printer via secondary Bluetooth port.

Connect printer

- Refresh list - Displays new printer list with discovering.
- Last connected - KDC connects to last connected printer.
- Device 1 to Device 10 - Displays up to 10 devices and KDC connects to selected printer.

Send data to printer

iOS SDK displays files which are located in application document directory and sends the contents of selected file to printer via KDC. Files can be downloaded into document directory through iTunes.

Commands Set

KDC400 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 KDC400 before sending individual commands. KDC400 would return “@” once wake up from sleep mode.

Command		Input	Output	Description	
0x0F0x4B0x03		None		Lock KDC commands	
0x0F0x4B0x02		None		Unlock KDC commands	
B		None	<i>value @</i>	Returns current battery capacity(<i>value</i> = 0~100)	
b	SC	<value>#	@ or !	Connect to printer	
	SD	<length>#<data>	@ or !	Send data to printer	
	SN	None	<i>Value @</i>	Discover printers and send list	
	SS	None	<i>value @</i>	Get printer connection status	
	T0	None	<i>value @</i>	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	@	Enable <i>Bluetooth</i> power off message to host
T9	None	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	0	@	Set Bluetooth ConnectDevice to SPP
	1	@	Set Bluetooth ConnectDevice to HID iOS
	2	@	Set Bluetooth ConnectDevice to iPhone
	3	@	Set Bluetooth ConnectDevice to SPP2.0
	4	@	Set Bluetooth ConnectDevice to HID normal
	G	@	Return ConnectDevice mode
TC	None	@	Connect to predefined <i>Bluetooth</i> MAC address and PIN code
TD	<value>#	@ or !	Set Bluetooth MAC address

	TE	<value>#	@ or !	Set PIN code. No PIN code if <value> is Null.
	TO	0	value@	Return current <i>Bluetooth</i> auto power on delay time
		1 value#	@	Set <i>Bluetooth</i> auto power on delay time
	TS	0	@	Disable ConfirmToSend
		1	@	Enable ConfirmToSend
	TV	<value>	@	Return KDC <i>Bluetooth</i> firmware version
	TW0	None	@	Disable leading null bytes
	TW1	None	@	Enable leading null bytes
	C	YMDHmS	@	Setting new date and time of KDC400
	c	None	YMDHmS @	Returns current date and time of KDC400
	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	B	0	@	Unlock buttons for normal operation
		1	@	Lock buttons to prevent undesired accidental operation
	b	0	@	Disable beep sound
		1	@	Enable beep sound
		2	@	Get beep sound setting
		3	@	Get beep volume setting
		V0	@	Low beep volume
		V1	@	High beep volume
		C	0	@

	1	@	Enable <i>Bluetooth</i> , Serial and USB connection status display
D	0	@	Unlock date and time setting
	1	@	Lock date and time setting
E	SP<count>#<string>	@	Set prefix string up to 11 characters
	SS<count>#<string>	@	Set suffix string up to 11 characters
	SA<value>#	@	Add/Remove AIM ID
	GP	value @	Get prefix string
	GS	value @	Get suffix string
	GA	value @	Get AIM ID position
	SO<value>#	@	Set start position of partial data
	GO	value @	Get start position of partial data
	SL<value>#	@	Set length of partial data
	GL	value @	Get length of partial data
	ST<value>	@	Set partial display action flag
	GT	value @	Get partial display action flag
HKG	None	value @	Get HID keyboard layout
HKS	<value>#	@	Set HID keyboard layout
H	TG	value @	Get HID Autolock time
	TS<value>#	@	Set HID Autolock time
I	MS0	@	Disable image capture
	MS1	@	Enable image capture
	MG	value@	Get current image capture mode 0 = Disabled 1 = Enabled
	NH	@	Snap and ship image
	FS<value>#	@	Set Image format 0x06 = JPEG (default) 0x08 = BMP

	FG	<i>value @</i>	Get current image format 0x06 = JPEG (default) 0x08 = BMP
	DS<value>#	@	Set pixel depth for BMP format 0x01 = 1 bit per pixel (B&W) 0x08 = 8 bits per pixel (Grayscale)
	DG	<i>value @</i>	Get current pixel depth 0x01 = 1 bit per pixel (B&W) 0x08 = 8 bits per pixel (Grayscale)
K	G	<i>value @</i>	Get current KDC Mode 0 = Normal Mode 1 = Application Mode
	S<value#>	@	Set KDC Mode 0 = Normal Mode 1 = Application Mode
L	0	@	Disable Display Scroll
	1	@	Enable Display Scroll
M	<value ; value>#	@ or !	Set start display position of message from Host
	B0	@	Failure Alert Beep
	B1	@	Success Alert Beep
	BC<val>#<val>#<val>#	@	Custom beep
	C0	@	Clear Screen
	d<value>#	@	Display bitmap without clear screen
	D<value>#	@	Display bitmap after clear screen
	f<value>#	@	Set message font size
	S<value>#	@ or !	Set message display duration from Host
	T<value><CR>	@	Display message from Host on KDC
nC	G	<i>value @</i>	Return current setting of HID Control Char

	S<value>#	@	Set HID Control Char
nd	BG	<i>value @</i>	Get HID barcode initial delay
	BS<value>#	@	Set HID barcode initial delay
	CG	<i>value @</i>	Get HID inter character delay
	CS<value>#	@	Set HID inter character delay
nD	G	<i>value @</i>	Get duplicate check setting in normal mode
	S0	@	Disable duplicate check in normal mode
	S1	@	Enable duplicate check in normal mode
nE	G	<i>value @</i>	Get auto erase setting
	S0	@	Disable auto erase
	S1	@	Enable auto erase
nF		<i>value @</i>	Get current data format
nMB	G	<i>value @</i>	Get the status of beep on MSR Data Error
	S<value>#	@	Set the status of beep on MSR Data Error
nMD	G	<i>value @</i>	Get MSR Data type
	S<value>#	@	Set MSR Data Type
nME	G	<i>value @</i>	Get MSR Encrypt mode
	S<value>#	@	Set MSR Data Encrypt mode
nMk	G	<i>value @</i>	Get AES Key length
	S<count>#<data>	@	Set AES Key length
nMK	G	<i>value @</i>	Get AES Key
	S<value>#	@	Set AES Key
nMS	G	<i>value @</i>	Get MSR track separator.
	S<value>#	@	Set MSR track separator.
nMT	G	<i>value @</i>	Get track selection
	S<value>#	@	Set track selection
nS1	None	@	Get available flash memory size
nT	G	<i>value @</i>	Get sleep timeout

	<i>S<value>#</i>	<i>@ or !</i>	Set sleep timeout
pm	0	@	Unlock partial display menu entry
	1	@	Lock partial display menu entry
Ps	<i>value#</i>	@	Set partial data display start position
pl	<i>value#</i>	@	Set partial data display length
pt	<i>value</i>	@	Set partial display action flag
S	0	@	Start Synchronization Mode
	1	@	Finish Synchronization Mode
T	<i>SD<value>#</i>	<i>@ or !</i>	Set Auto Trigger Reread delay time
	<i>GS</i>	<i>value @</i>	Get Auto Trigger Reread delay time
	<i>SM0</i>	@	Disable Auto Trigger Mode
	<i>SM1</i>	@	Enable Auto Trigger Mode
	<i>GM</i>	<i>value @</i>	Get Auto Trigger Mode
TS	<i>Value#</i>	<i>@ or !</i>	Set termination character after barcode data if data format is barcode only
TG	None	<i>value @</i>	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
Y	0	@	Display scanned barcode data, barcode type, date and time
	1	@	Display scanned barcode data, date and time, battery status
	2	@	Display scanned barcode data, barcode type, battery status
	3	@	Display scanned barcode data and the number of stored barcodes.
	5	@	Display Barcode Only

	Z	SS<string>	@	Set secure token
		SG	string@	Return secure token
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 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 KDC400 from sleep mode
w		value #	@ or !	Select wedging data format (Barcode only, packet data)
Z		value #	@ or !	Set security level
z		None	value @	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 KDC400

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

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

'bSC' (Connect to printer)

Parameters: Printer MAC address

Format: 'bSC<MAC address>#', where MAC address is 12 characters. KDC connects to previous connected device if <MAC address> is empty

Output: '@' or '!', where '@' means connection is succeed and '!' means connection is failed

Purpose: Connect KDC400 to Bluetooth printer

[Note] This command is only applicable to KDC415, KDC425 and KDC430 model.

'bSD' (Send data to printer)

Parameters: Data length and data

Format: 'bSD<length>#<data>', where length is maximum 4 bytes.

Output: '@' or '!', where '@' means data is transmitted successfully and '!' means data is failed to transmit.

Purpose: Send data to printer

[Note] This command is only applicable to KDC415, KDC425 and KDC430 model.

'bSN' (Discover printers and send list)

Parameters: None

Format: 'bSN'

Output: '<value>@', where value is either '!' or the list of MAC address and friendly name of available devices in following format

- Start byte(1): 0x04
- Device Info(n x 16): Each device info is 16 bytes long as below and it is repeated.
 - Mac address(12): Each device has 12 bytes Mac address. Sends 'n' devices MAC address continuously.
 - End of mac address(1): 0x80
 - PIN code terminator(1): 0xAA
 - Prefix terminator(1): 0xAA
 - Suffix terminator(1): 0xAA
- End of mac address(1): 0xFE
- Friendly name(n x m): Each device has 'm' bytes (maximum 13 characters) friendly name and separated with byte 0xAA
- End of data(2 bytes): 0xFE 0xFE

Purpose: Discover printers and send list

[Example]

- MAC address: 001901205A9A, Friendly name: KDC300[000100
- value is
04 30 30 31 39 30 31 32 30 35 41 39 41 80 AA AA AA FE 4B 44 43 33 30 30 5B 30
30 30 31 30 30 AA FE FE

[Note] This command is only applicable to KDC415, KDC425 and KDC430 model.

'bSS' (Get printer connection status)

Parameters: None

Format: 'bSS'

Output: '<value>@', where

value	Hex No.	Connection Status
0	0x00	Not Connected
1	0x01	Connected

Purpose: Get printer connection status

[Note] This command is only applicable to KDC415, KDC425 and KDC430 model.

'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 : HID button enabled(1)/disabled(0)

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)

bit 7 : SendLeadingNullByte 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<value>', where <value> is 0 or 1

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

Output: '@'

Purpose: Turns on(1)/off(0) KDC400 *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: KDC400 starts pairing mode

'bT3' (*Bluetooth* auto connection)

Parameters: 0 or 1

Format: 'bT3<value>', where <value> 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 KDC400 is connected to the HOST

Purpose: Enables(1)/Disables(0) KDC400 to try connection to *Bluetooth* HOST automatically after power on

[Note] KDC400 auto connection mode can't be changed to "Disable" from "Enable" while KDC400 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<value>', where <value> is 0 or 1

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

Output: '@'

Purpose: Enables(1)/Disables(0) KDC400 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) KDC400 to turn *Bluetooth* module power off when KDC400 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<value>', where <value> is 0 or 1

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

Output: '@'

Purpose: Enables(1)/Disables(0) KDC400 to beep on every 15 minutes when *Bluetooth* power is on and *Bluetooth* is disconnected, but Auto power off option is disabled. KDC400 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

Auto power off timeout value	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: KDC400 can send *Bluetooth* power off message "BTOFF" to host upon executing "bT10" *Bluetooth* power off command. KDC400 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: '@'

Purpose: KDC400 can send *Bluetooth* power off message "BTOFF" to host upon executing "bT10" *Bluetooth* power off command. KDC400 would send this "BTOFF" message if user enables *Bluetooth* power off message using "bT81" command. The default is disable *Bluetooth* power off message.

'bT9' (Get *Bluetooth* MAC address)

Parameters: None

Format: 'bT9'

Output: '<value>@', where <value> is the MAC address of KDC400 *Bluetooth* module and @ means the end of <value>

Purpose: To get current *Bluetooth* MAC address of KDC400 *Bluetooth* module

'bTA0' (Set KDC400 in *Bluetooth* Slave Mode)

Parameters: None

Format: 'bTA0'

Output: '@'

Purpose: KDC400 can be configured to *Bluetooth* master or slave Mode. 'bTA0' command would set KDC400 to *Bluetooth* slave mode. The default is slave mode.

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

'bTA1' (Set KDC400 in *Bluetooth* Master Mode)

Parameters: None

Format: 'bTA1'

Output: '@'

Purpose: KDC400 can be configured to *Bluetooth* master or slave Mode. 'bTA1' command would set KDC400 to *Bluetooth* master mode. The default is slave mode.

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

'bTc0' (Set Bluetooth ConnectDevice to SPP)

Parameters: None

Format: 'bTc0'

Output: '@'

Purpose: This command enables KDC *Bluetooth* profile to *Bluetooth* Spec2.1+EDR SPP.

Note: User should select SPP2.0 if the host is not compatible with *Bluetooth* Spec2.1+EDR.

'bTc1' (Set Bluetooth ConnectDevice to HID iOS)

Parameters: None

Format: 'bTc1'

Output: '@'

Purpose: This command enables KDC *Bluetooth* profile to HID iOS. HID iOS profile can be used with devices supporting HID profile such as iPhone 4.0 and iPad.

Note: This command is only applicable to KDC400(i)/300(i) with Bluetooth firmware version 1.2.2rt or higher.

'bTc2' (Set Bluetooth ConnectDevice to iPhone)

Parameters: None

Format: 'bTc2'

Output: '@'

Purpose: This command enables KDC *Bluetooth* profile to Made for iPhone *Bluetooth*.

Note: This command is only applicable to KDC400i and KDC400i models.

'bTc3' (Set Bluetooth ConnectDevice to SPP2.0)

Parameters: None

Format: 'bTc3'

Output: '@'

Purpose: This command enables KDC *Bluetooth* profile to *Bluetooth Spec2.0+EDR SPP*.

Note: User should select SPP2.0 if the host is not compatible with *Bluetooth Spec2.1+EDR*.

'bTc4' (Set ConnectDevice to HID normal)

Parameters: None

Format: 'bTc4'

Output: '@'

Purpose: This command enables KDC *Bluetooth* profile to HID normal.

Note: User should select HID normal if the host is not Apple device.

'bTcG' (Return ConnectDevice mode)

Parameters: None

Format: 'bTcG'

Output: '<value>@', where <value> is the selected *Bluetooth* profile and @ means the end of <value>.

Value	Hex. Number	Bluetooth Profile
0	0x00	SPP
1	0x01	HID
2	0x02	IPHONE
3	0x03	SPP2.0
4	0x04	HID normal

Purpose: To get current *Bluetooth* profile mode

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

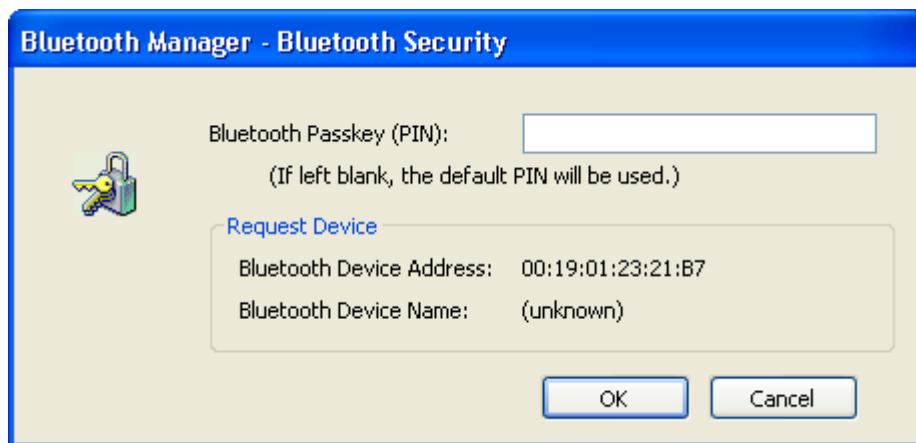
Parameters: None

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 KDC400 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 KDC400 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

MAC Address	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 KDC400 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

PIN Code	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 KDC400 as master mode using "bTA" command depends on corresponding device *Bluetooth* stack.

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

'bTO0' (Read *Bluetooth* auto power on delay time)

Parameters: None

Format: 'bTO0'

Output: '<value>@', where <value> is the delay time (seconds) of *Bluetooth* auto power on in hexadecimal and @ means the end of <value>

Purpose: To get current *Bluetooth* auto power on delay time

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

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

'bTO1' (Set *Bluetooth* auto power on delay time)

Parameters: Delay time of *Bluetooth* auto power on

Format: 'bTO1<value>#' where *value* is between 0 and 10 seconds

Auto power on delay time	Character	Hex. Number
0~9	"0" ~ "9"	30 ~ 39
10	"A"	41

Output: '@'

Purpose: Setting current *Bluetooth* auto power on delay time from 0 to 10 seconds. 0 means that auto power on delay is disabled.

[Note] It will be set to 0 if *value* is less than 0; set to 10 seconds if *value* is greater than 10. Default values is 0(disabled).

[Example] Send 7 bytes data (0x62, 0x54, 0x4F, 0x31, 0x33, 0x23) to set *Bluetooth* auto power on delay time to 3 seconds

- 0x62: ASCII code of character "b" in hexadecimal
- 0x54: ASCII code of character "T" in hexadecimal
- 0x4F: ASCII code of character "O" in hexadecimal
- 0x31: ASCII code of character "1" in hexadecimal
- 0x33: ASCII code of character "3" in hexadecimal
- 0x23: ASCII code of character "#" in hexadecimal
- bTO13#

'bTS0' (Disable ConfirmToSend)

Parameters: None

Format: 'bTS0'

Output: '@'

Purpose: Disable ConfirmToSend option. KDC doesn't send barcode data to the host until host send "AC" command to KDC if ConfirmToSend is enabled. The default is enabled.

'bTS1' (Enable ConfirmToSend)

Parameters: None

Format: 'bTS1'

Output: '@'

Purpose: Enable ConfirmToSend option. KDC doesn't send barcode data to the host until host send "AC" command to KDC if ConfirmToSend is enabled. The default is enabled.

[Note] KDC resumes barcode data transmission though "AC" command is not received if KDC enters sleep mode.

'bTV' (*Bluetooth* Firmware Version)

Parameters: None

Format: 'bTV'

Output: '<value>@', where <value> is the *Bluetooth* firmware version and @ means the end of <value>

Purpose: To get current *Bluetooth* firmware version

'bTW0' (Disable Leading Null bytes)

Parameters: None

Format: 'bTW0'

Output: @

Purpose: KDC can send out three leading null bytes to wake up connected *Bluetooth* device. This command disables sending three leading null bytes.

'bTW1' (Enable Leading Null bytes)

Parameters: None

Format: 'bTW1'

Output: @

Purpose: KDC can send out three leading null bytes to wake up connected *Bluetooth* device. This command enables sending three leading null bytes.

'C' (Set date and time)

Parameters: Date and Time value to be set in KDC400

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 KDC400

[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 KDC400

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

- 0x00: Year 2000
- 0x02: February
- 0x03: 3rd
- 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: '!' – KDC400 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 KDC400 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 KDC400 up, down and scan buttons for normal operation

'GB1' (Lock button)

Parameters: None

Format: 'GB1'

Output: '@'

Purpose: To lock KDC400 up, down and scan buttons. Locking buttons prevents undesired accidental operation and keeps KDC400 in sleep mode (battery saving mode)

'Gb0' (Beeper off)

Parameters: None

Format: 'Gb0'

Output: '@'

Purpose: Disable KDC400 beep sound

'Gb1' (Beeper on)

Parameters: None

Format: 'Gb1'

Output: '@'

Purpose: To enable KDC400 beep sound

'Gb2' (Get Beep sound setting)

Parameters: None

Format: 'Gb2'

Output: '<value>@', where value is

value	Hex No.	Note
0	0x00	Disabled
1	0x01	Enabled

Purpose: Get KDC beep sound setting

'Gb3' (Get Beep volume setting)

Parameters: None

Format: 'Gb3'

Output: '<value>@', where value is

value	Hex No.	Note
0	0x00	Low volume
1	0x01	High volume

Purpose: Get KDC beep volume setting

'GbV0' (Low beep volume)

Parameters: None

Format: 'GbV0'

Output: '@'

Purpose: Set beep volume to low

'GbV1' (High beep volume)

Parameters: None

Format: 'GbV1'

Output: '@'

Purpose: Set beep volume to high

'GC0' (Disable port connection status display)

Parameters: None

Format: 'GC0'

Output: '@'

Purpose: Disable *Bluetooth*, Serial and USB connection status display. KDC400 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. KDC400 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 KDC400 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 KDC400 menu.

'GESP' (Set Prefix String)

Parameters: Prefix count and string

Format: 'GESP<count>#<string>', where "1" ≤ *count* ≤ "B" and *string* is a series of prefix characters up to 11.

count	Character	Hex No.
1~9	"1" ~ "9"	31 ~ 39
10~11	"A" ~ "B"	41 ~ 42

Output: '@'

Purpose: Set prefix string up to 11 characters.

'GESS' (Set Suffix String)

Parameters: count and string

Format: 'GESS<count>#<string>', where "1" ≤ *count* ≤ "B" and *string* is a series of suffix characters up to 11.

count	Character	Hex No.
1~9	"1" ~ "9"	31 ~ 39
10~11	"A" ~ "B"	41 ~ 42

Output: '@'

Purpose: Set suffix string up to 11 characters.

'GESA' (Add or Remove AIM ID)

Parameters: AIM ID Position

Format: 'GESA<value>#', where "0" ≤ value ≤ "2"

- No AIM ID if value = "0"
- AIM ID at the end of prefix if value = "1"
- AIM ID at the end of suffix if value = "2"

Value	Character	Hex No.
0~2	"0" ~ "2"	30 ~ 32

Output: '@'

Purpose: Add or Remove AIM ID

'GEGP' (Get Prefix String)

Parameters: string

Format: 'GEGP'

Output: '<string>@', where <string> is a series of prefix characters up to 11.

Purpose: Get prefix string

'GEGS' (Get Suffix String)

Parameters: string

Format: 'GEGS'

Output: '<string>@', where <string> is a series of suffix characters up to 11.

Purpose: Get suffix string

'GEGA' (Get AIM ID Setting)

Parameters: value

Format: 'GEGA'

Output: '<value>@', where $0 \leq \text{value} \leq 2$

- No AIM ID if value = 0
- AIM ID at the end of prefix if value = 1
- AIM ID at the end of suffix if value = 2

[Note] <value> is numeric value, not character "0"(0x30)~"2"(0x32)

Purpose: Get AIM ID Setting

'GESO' (Set Start Position of Partial Data)

Parameters: Partial data start position

Format: 'GESO<value>#', where "1" ≤ value ≤ "3E7"

Value	Character	Hex No.
1~9	"1" ~ "9"	31 ~ 39
...
42~47	"2", "A" ~ "2", "F"	32,41 ~ 32,46
...
999	"3", "E", "7"	33,45,37

Output: '@'

Purpose: Set Start Position of Partial Data

'GEGO' (Get Start Position of Partial Data)

Parameters: Partial data start position

Format: 'GEGO'

Output: '<value>@', where $1 \leq \text{value} \leq 999$

Purpose: Get Start Position of Partial Data

'GESL' (Set Length of Partial Data)

Parameters: Partial data length

Format: 'GESL<value>#', where "0" ≤ value ≤ "3E7". "0" means no partial data option is enabled.

Value	Character	Hex No.
1~9	"1" ~ "9"	31 ~ 39
...
42~47	"2", "A" ~ "2", "F"	32,41 ~ 32,46
...
999	"3", "E", "7"	33,45,37

Output: '@'

Purpose: Set Length of Partial Data

'GEGE' (Get Length of Partial Data)

Parameters: Partial data length

Format: 'GEGE'

Output: '<value>@', where $0 \leq \text{value} \leq 999$. "0" means no partial data option is enabled.

Purpose: Get Length of Partial Data

'GEST' (Set Partial Data Action Flag)

Parameters: "0" or "1" in character. 0x30 or 0x31 in hexadecimal number

Format: 'GEST<value>', where value = "0" or "1". "0" means Erase. "1" means Select.

"0" or "1" in character. 0x30 or 0x31 in hexadecimal number

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

Output: '@'

Purpose: Set partial data action

'GEGT' (Get Partial Data Action Flag)

Parameters: None

Format: 'GEGT'

Output: '<value>@', where value = "0" or "1". "0" means Erase. "1" means Select.
"0" or "1" in character. 0x30 or 0x31 in hexadecimal number

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

Purpose: Get partial data action flag

'GHKG' (Get HID Keyboard layout)

Parameters: None

Format: "GHKG"

Output: '<value>@', where value is

value	Character	Hex No.	Note
0	"0"	0x00	English
1	"1"	0x01	German
2	"2"	0x02	French
3	"3"	0x03	Italian
4	"4"	0x04	Spanish

Purpose: Get HID keyboard layout mode

'GHKS' (Set HID Keyboard layout)

Parameters: HID keyboard layout value

Format: "GHKS<value>#", where value is

value	Character	Hex No.	Note
0	"0"	0x30	English
1	"1"	0x31	German
2	"2"	0x32	French
3	"3"	0x33	Italian
4	"4"	0x34	Spanish

Output: '@'

Purpose: Set HID keyboard layout

'GHTG' (Get HID Autolock Time)

Parameters: None

Format: 'GHTG'

Output: '<value>@', where value = 0,1,2,3,4,5,10, 15 minutes in hexadecimal

Value	Hex No.
0	0x0
1	0x1
2	0x2
3	0x3
4	0x4
5	0x5
A	0xA
F	0xF

Purpose: Get HID Autolock Time in minutes

[Note] value 0 means Autolock is disabled. The unit of value is minute.

'GHTS' (Set HID Autolock Time)

Parameters: HID autolock time in minute

Format: 'GHTS<value>#', where value is the HID autolock time to set in hexadecimal minutes.

Value	Character	Hex No.
0	"0"	30
1	"1"	31
2	"2"	32
3	"3"	33
4	"4"	34
5	"5"	35
A	"A"	41
F	"F"	46

Output: '@'

Purpose: Set HID Autolock Time in minutes

[Note] value 0 means Autolock is disabled. The unit of value is minute.

'GIDG' (Get current pixel depth)

Parameters: None

Format: 'GIDG'

Output: '<value>@', where value = 1 or 8 in hexadecimal.

Value	Hex No.
1	0x01
8	0x08

Purpose: Get current pixel depth.

[Note] This command is only applicable to BMP format.

[Note] This command is only applicable to KDC420 and KDC425 model.

'GIDS' (Set current pixel depth of BMP format)

Parameters: BMP format pixel depth (B&W or Grayscale)

Format: 'GIDS<value>#', where value=1 if 1 bit black and white image and value=8 if 8bits grayscale image.

Value	Character	Hex No.
1	"1"	31
8	"8"	38

Output: '@'

Purpose: Set the current BMP format pixel depth.

[Note] This command is only applicable to KDC420 and KDC425 model.

'GIFG' (Get current image format)

Parameters: None

Format: 'GIFG'

Output: '<value>@', where value=6 if JPEG and value=8 if BMP.

Value	Hex No.
6	0x06
8	0x08

Purpose: Get the current KDC400 image format

[Note] This command is only applicable to KDC420 and KDC425 model.

'GIFS' (Set current image format)

Parameters: Image format (JPEG or BMP)

Format: 'GIFS<value>#', where value=6 if JPEG and value=8 if BMP.

Value	Character	Hex No.
6	"6"	36
8	"8"	38

Output: '@'

Purpose: Set the current KDC400 image format

[Note] This command is only applicable to KDC420 and KDC425 model.

'GIMS0' (Disable image capture)

Parameters: None

Format: 'GIMS0'

Output: '@'

Purpose: Disable KDC400 image capture option

[Note] This command is only applicable to KDC420 and KDC425 model.

'GIMS1' (Enable image capture)

Parameters: None

Format: 'GIMS1'

Output: '@'

Purpose: Enable KDC400 image capture option

[Note] This command is only applicable to KDC420 and KDC425 model.

'GIMG' (Get image capture setting)

Parameters: None

Format: 'GIMG<value>@', where value = 0 if disabled, value=1 if enabled.

Value	Hex No.
0	0x00
1	0x01

Output: '@'

Purpose: Enable KDC400 image capture option

[Note] This command is only applicable to KDC420 and KDC425 model.

'GINH' (Snap and ship image)

Parameters: None

Format: 'GINH'

Output: '@'

Purpose: Capture image

[Note] This command is only applicable to KDC420 and KDC425 model.

'GKG' (Get current KDC Mode)

Parameters: None

Format: "GKG"

Output: '<value>@', where value = 0 (0x00 in Hex) if Normal mode; value = 1 (0x01 in Hex) if Application mode

Value	Hex No.
0	0x00
1	0x01

Purpose: Get current KDC Mode

'GKS' (Set current KDC Mode)

Parameters: KDC Mode setting value

Format: "GKS<value>#", where value = 0 for Normal mode and value = 1 for Application mode

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

Output: '@'

Purpose: Set current KDC Mode

'GL0' (Disable Display Scroll)

Parameters: None

Format: "GL0"

Output: '@'

Purpose: Disable display scroll

'GL1' (Enable Display Scroll)

Parameters: None

Format: "GL1"

Output: '@'

Purpose: Enable display scroll

'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 1st row and the 1st 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

'GMBC' (Custom beep tone)

Parameters: On time, Off time, Repeat number

Format: 'GMBC<value1>#<value2>#<value3>#', where value1 is On time in hexadecimal millisecond, value2 is Off time in hexadecimal millisecond and value3 is the number of repeat in hexadecimal.

msec	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: '@'

Purpose: Play custom beep tone

[Example] Send following string to play beep tone with 100msec on time, 200msec off time and repeat 10 times.

- 0x47: ASCII code of character "G" in hexadecimal
- 0x4D: ASCII code of character "M" in hexadecimal
- 0x42: ASCII code of character "B" in hexadecimal
- 0x43: ASCII code of character "C" in hexadecimal
- 0x36: ASCII code of character "6" in hexadecimal
- 0x34: ASCII code of character "4" in hexadecimal
- 0x23: ASCII code of character "#" in hexadecimal
- 0x43: ASCII code of character "C" in hexadecimal
- 0x38: ASCII code of character "8" in hexadecimal
- 0x23: ASCII code of character "#" in hexadecimal
- 0x41: ASCII code of character "A" in hexadecimal
- 0x23: ASCII code of character "#" in hexadecimal

'GMC0' (Clear screen)

Parameters: None

Format: 'GMC0'

Output: '@'

Purpose: Clear KDC screen

'GMd' (Display bitmap without clear screen)

Parameters: bitmap index

Format: 'GMd<value>#', where value is the bitmap index between 1 and 50 in hexadecimal

value	Character	Hex No.
1	"1"	31
2	"2"	32
...
50	"3", "2"	0x33,0x32

Output: '@'

Purpose: Display bitmap without clear screen

Note: This command is only supported by KDC firmware 3.0 or above.

'GMD' (Display bitmap after clear screen)

Parameters: bitmap index

Format: 'GMD<value>#', where value is the bitmap index between 1 and 50 in hexadecimal

value	Character	Hex No.
1	"1"	31
2	"2"	32
...
50	"3", "2"	0x33,0x32

Output: '@'

Purpose: Display bitmap after clear screen

Note: This command is only supported by KDC firmware 3.0 or above.

'GMf' (Set message font size)

Parameters: Font size

Format: 'GMf<value>#', where

value	Character	Hex No.	Font Size
0	"0"	30	8 x 8
1	"1"	31	10 x 16
2	"2"	32	16 x 16
3	"3"	33	16 x 24
4	"4"	34	16 x 32
5	"5"	35	24 x 24
6	"6"	36	24 x 32
7	"7"	37	32 x 32

Output: '@'

Purpose: Set KDC screen message font size

[Example] Send following string to display 8x8 font size.

- 0x47: ASCII code of character "G" in hexadecimal
- 0x4D: ASCII code of character "M" in hexadecimal
- 0x42: ASCII code of character "f" in hexadecimal
- 0x30: ASCII code of character "0" in hexadecimal
- 0x23: ASCII code of character "#" in hexadecimal

'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>

'GnCG' (Get HID control character mode)

Parameters: None

Format: 'GnCG'

Output: '<value>@', where value is the HID control character mode.

value	Hex No.	Mode
0	0x00	Disabled
1	0x01	Alt+Numpad
2	0x02	^+Character

Purpose: Get HID control character mode

'GnCS' (Set HID Control Character)

Parameters: HID control character mode

Format: 'GnCS<value>#', where value is the HID control character mode

value	Hex No.	Mode
'0'	0x30	Disabled
'1'	0x31	Alt+Numpad
'2'	0x32	^+Character

Output: '@'

Purpose: Set HID control character mode

'GndBG' (Get HID initial delay)

Parameters: None

Format: 'GndBG'

Output: '<value>@', where value is the HID initial delay in second(s).

value	Hex No.
0	0x00
1	0x01
2	0x02
3	0x03
5	0x05
10	0x0A

Purpose: Get HID initial delay

'GndBS' (Set HID initial delay)

Parameters: Initial delay in msec

Format: 'GndBS<value>#', where value is the HID initial delay in second(s)

value	Hex No.
'0'	0x30
'1'	0x31
'2'	0x32
'3'	0x33
'5'	0x35
'A'	0x41

Output: '@'

Purpose: Set HID initial delay

'GndCG' (Get HID inter-character delay)

Parameters: None

Format: 'GndCG'

Output: '<value>@', where value is the HID inter-character delay in msec.

value	Hex No.
0	0x00
10	0x0A
20	0x14
30	0x1E
50	0x32
100	0x64

Purpose: Get HID inter-character delay

'GndCS' (Set HID inter-character delay)

Parameters: Inter-character delay in msec

Format: 'GndCS<value>#', where value is the HID inter-character delay in msec

value	Hex No.
'0'	0x30
'A'	0x41
'1','4'	0x31, 0x34
'1','E'	0x31, 0x45
'3','2'	0x33, 0x32
'6','4'	0x36, 0x34

Output: '@'

Purpose: Set HID inter-character delay

'GnDG' (Get duplicate check setting in normal mode)

Parameters: None

Format: 'GnDG'

Output: '<value>@', where value 0 means disabled and 1 means enabled.

value	Hex No.
0	0x00
1	0x01

Purpose: Get duplicate check setting in normal mode

'GnDS0' (Disable duplicate check in normal mode)

Parameters: None

Format: 'GnDS0'

Output: '@'

Purpose: Disable duplicate check in normal mode

'GnDS1' (Enable duplicate check in normal mode)

Parameters: None

Format: 'GnDS1'

Output: '@'

Purpose: Enable duplicate check in normal mode

'GnEG' (Get current auto erase setting of Auto Erase)

Parameters: None

Format: 'GnEG'

Output: '<value>@', where value 0 means disabled and 1 means enabled.

value	Hex No.
0	0x00
1	0x01

Purpose: Get current auto erase setting of auto erase mode

'GnES0' (Disable auto erase)

Parameters: None

Format: 'GnES0'

Output: '@'

Purpose: Disable auto erase

'GnES1' (Enable auto erase)

Parameters: None

Format: 'GnES1'

Output: '@'

Purpose: Enable auto erase

'GnF' (Get current Data Format)

Parameters: None

Format: 'GnF'

Output: '<value>@', where value 0 means Barcode only and 1 means Packet data

value	Hex No.
0	0x00
1	0x01

Purpose: Get current data format

'GnMBG' (Get MSR error beep tone)

Parameters: None

Format: 'GnMBG'

Output: '<value>@', where value is the MSR error beep tone.

value	Hex No.	MSR error beep tone
0	0x00	No beep on MSR data error
1	0x01	Beep on MSR data error

Purpose: Get MSR error beep tone

[Note] This command is only applicable to KDC415, KDC425 and KDC430 model.

'GnMBS' (Set MSR error beep tone)

Parameters: MSR data type

Format: 'GnMBS<value>#', where value is the MSR error beep tone.

value	Hex No.	MSR error beep tone
'0'	0x30	No beep on MSR data error
'1'	0x31	Beep on MSR data error

Output: '@'

Purpose: Set MSR error beep tone

[Note] This command is only applicable to KDC415, KDC425 and KDC430 model.

'GnMDG' (Get MSR Data Type)

Parameters: None

Format: 'GnMDG'

Output: '<value>@', where value is the MSR data type

value	Hex No.	MSR data type
0	0x00	Payload only
1	0x01	Packet data

Purpose: Get MSR data type

[Note] This command is only applicable to KDC415, KDC425 and KDC430 model.

'GnMDS' (Set MSR Data Type)

Parameters: MSR data type

Format: 'GnMDS<value>#', where value is the MSR data type.

value	Hex No.	MSR data type
'0'	0x30	Payload only
'1'	0x31	Packet data

Output: '@'

Purpose: Set MSR data type

[Note] This command is only applicable to KDC415, KDC425 and KDC430 model.

'GnMEG' (Get MSR Data Encrypt Mode)

Parameters: None

Format: 'GnMEG'

Output: '<value>@', where value is the MSR data encrypt mode

value	Hex No.	MSR encrypt mode
0	0x00	No encryption
1	0x01	AES

Purpose: Get MSR data encrypt mode

[Note] This command is only applicable to KDC415, KDC425 and KDC430 model.

'GnMES' (Set MSR data encrypt mode)

Parameters: MSR data encrypt mode

Format: 'GnMES<value>#', where value is the MSR data encrypt mode.

value	Hex No.	MSR data encrypt mode
'0'	0x30	No encryption
'1'	0x31	AES

Output: '@'

Purpose: Set MSR data encrypt mode

[Note] This command is only applicable to KDC415, KDC425 and KDC430 model.

'GnMES' (Set MSR data encrypt mode)

Parameters: MSR data encrypt mode

Format: 'GnMES<value>#', where value is the MSR data encrypt mode.

value	Hex No.	MSR data encrypt mode
'0'	0x30	No encryption
'1'	0x31	AES

Output: '@'

Purpose: Set MSR data encrypt mode

[Note] This command is only applicable to KDC415, KDC425 and KDC430 model.

'GnMkG' (Get AES Key length)

Parameters: None

Format: 'GnMkG'

Output: '<value>@', where value is the AES Key length.

value	Hex No.	AES Key length
0	0x00	128 bits
1	0x01	192 bits
2	0x02	256 bits

Purpose: Get AES Key length

[Note] This command is only applicable to KDC415, KDC425 and KDC430 model.

'GnMkS' (Set AES Key length)

Parameters: AES Key length

Format: 'GnMkS<value>#', where value is the AES key length

value	Hex No.	AES Key length
0	0x30	128 bits
1	0x31	192 bits
2	0x32	256 bits

Output: '@'

Purpose: Get AES Key length

[Note] This command is only applicable to KDC415, KDC425 and KDC430 model.

'GnMKG' (Get AES Key)

Parameters: None

Format: 'GnMKG'

Output: '<value>@', where value is the AES Key string.

Purpose: Get AES Key

[Note] This command is only applicable to KDC415, KDC425 and KDC430 model.

'GnMKS' (Set AES Key)

Parameters: AES Key up to 32 characters

Format: 'GnMKS<count>#<data>', where count is the length of <data> string in hexadecimal

Output: '@'

Purpose: Get AES Key length

[Note] This command is only applicable to KDC415, KDC425 and KDC430 model.

'GnMSS' (Set MSR track separator)

Parameters: MSR track separator

Format: 'GnMSS<value>#', where value is the MSR track separator.

value	Hex No.	MSR track separator
'0'	0x30	None
'1'	0x31	Space
'2'	0x32	Comma
'3'	0x33	Semi Colon
'4'	0x34	CR
'5'	0x35	LF
'6'	0x36	CR&LF
'7'	0x37	Tab

Output: '@'

Purpose: Set MSR track separator

[Note] This command is only applicable to KDC415, KDC425 and KDC430 model.

'GnMTG' (Get MSR track selection status)

Parameters: None

Format: 'GnMTG'

Output: '<value>@', where value is the MSR track selection.

Bit #	Value	MSR track selection
0	1 (0x01)	Track1 is selected
1	2(0x02)	Track2 is selected
2	4(0x04)	Track3 is selected

Purpose: Get MSR track selection

[Note] This command is only applicable to KDC415, KDC425 and KDC430 model.

'GnMTS' (Set MSR track selection)

Parameters: MSR track selection

Format: 'GnMTS<value>#', where value is the MSR track selection.

Bit #	Value	MSR track selection
0	1 (0x01)	Track1 is selected
1	2(0x02)	Track2 is selected
2	4(0x04)	Track3 is selected

Output: '@'

Purpose: Set MSR track selection

[Note] This command is only applicable to KDC415, KDC425 and KDC430 model.

'GnS1' (Get available flash memory size)

Parameters: None

Format: 'GnS1'

Output: '<value>@', where value is the available flash memory size in hexadecimal.

Purpose: Get available flash memory size

'GnTG' (Get sleep timeout)

Parameters: None

Format: "GnTG"

Output: <value>@, where *value* is

value	Hex No.	Note
0	0x00	Disabled
1	0x01	1 second
2	0x02	2 seconds
3	0x03	3 seconds
4	0x04	4 seconds
5	0x05	5 seconds
10	0x0A	10 seconds
20	0x14	20 seconds
30	0x1E	30 seconds
60	0x3C	1 minute
120	0x78	2 minutes
300	0x12C	5 minutes
600	0x258	10 minutes

Purpose: Get KDC sleep timeout

'GnTS' (Set sleep timeout)

Parameters: sleep timeout in second

Format: "GnTS<value>#", where *value* is the sleep timeout in second

Sec	Character	Hex No.	Note
0	"0"	0x30	Disabled
1	"1"	0x31	1 second
2	"2"	0x32	2 seconds
3	"3"	0x33	3 seconds
4	"4"	0x34	4 seconds
5	"5"	0x35	5 seconds
10	"A"	0x41	10 seconds
20	"1","4"	0x31,0x34	20 seconds
30	"1","E"	0x31,0x45	30 seconds
60	"3","C"	0x33,0x43	1 minute
120	"7","8"	0x37,0x38	2 minutes
300	"1","2","C"	0x31,0x32,0x43	5 minutes
600	"2","5","8"	0x32,0x35,0x38	10 minutes

Output: '!' – Invalid parameter

'@' – Set sleep timeout successfully

Purpose: Set KDC sleep timeout

'Gpm0' (Unlock Partial data display menu entry)

Parameters:	None
Format:	"Gpm0"
Output:	'@' – all other times
Purpose:	KDC has partial data display option which can be locked to prevent unintentional setting change. This command allows user enter into partial data display menu in Scan Option menu.

'Gpm1' (Lock Partial data display menu entry)

Parameters:	None
Format:	"Gpm1"
Output:	'@' – all other times
Purpose:	KDC has partial data display option which can be locked to prevent unintentional setting change. This command prevents user enter into partial data display menu in Scan Option menu so that user can't change the start position and length of display character.

'Gps' (Partial data display start position)

Parameters: 1 to 256 which indicates the partial data display start position

Format: "<value>#" where <value> is the hexadecimal start position value and # means the end of <value>

s	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
.....
256	"1","0","0"	31,30,30

Output: '@' – all other times

Purpose: KDC has partial data display option which displays only part of scanned barcode. This command allows user to define the start position of partial display.

'Gpl' (Partial data display length)

Parameters: 0 to 39 which indicates the display length. KDC would display the entire scanned barcode if the length value is 0.

Format: "<value>#" where <value> is the hexadecimal display length value and # means the end of <value>

I	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

39	"2", "7"	32,37

Output: '@' – all other times

Purpose: KDC has partial data display option which displays only part of scanned barcode. This command allows user to define the length of partial display.

'Gpt' (Set Partial Display Action Flag)

Parameters: Partial display action flag

Format: "Gpt<value>", where value = "0" or "1". "0" means Erase. "1" means Select.

"0" or "1" in character. 0x30 or 0x31 in hexadecimal number

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

Output: '@'

Purpose: Set partial display action flag

'GS0' (Finish Synchronization)

Parameters: None

Format: 'GS0'

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: '<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

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

'GtGD' (Get Auto Trigger Reread Delay)

Parameters: None

Format: 'GtGD

Output: '<value>@', where <value> is between 0 and 4, and @ means the end of <value>

value	Hex No.
0	0x00
1	0x01
2	0x02
3	0x03
4	0x04

Purpose: To get auto trigger reread delay among continuous, short, medium, long and extra long

[Example] Send 4 bytes (0x47, 0x74, 0x47, 0x44) to get the auto trigger reread delay.

- 0x47: ASCII code of character "G"
- 0x74: ASCII code of character "t"
- 0x47: ASCII code of character "G"
- 0x44: ASCII code of character "D"

'GtGM' (Get Auto Trigger Mode)

Parameters: None

Format: 'GtGM'

Output: '<value>@', where <value> is between 0 and 1, and @ means the end of <value>

Value	Hex No.
0	0x00
1	0x01

Purpose: Disable auto trigger mode.

[Example] Send 4 bytes (0x47, 0x74, 0x47, 0x4D) to get the auto trigger mode

- 0x47: ASCII code of character "G"
- 0x74: ASCII code of character "t"
- 0x47: ASCII code of character "G"
- 0x4D: ASCII code of character "M"

'GtSD' (Set Auto Trigger Reread Delay)

Parameters: 0, 1, 2, 3 or 4 (0x00, 0x01, 0x02 or 0x03 in hexadecimal number)

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

value	Hex No.
0	00
1	01
2	02
3	03
4	04

Output: '!' – Invalid parameter

'@' – Set Reread delay successfully

Purpose: To set auto trigger reread delay among continuous, short, medium, long and extra long

[Example] Send 6 bytes (0x47, 0x74, 0x53, 0x44, 0x00, 0x23) to set the auto trigger reread delay to continuous.

- 0x47: ASCII code of character "G"
- 0x74: ASCII code of character "t"
- 0x53: ASCII code of character "S"
- 0x44: ASCII code of character "D"
- 0x00: ASCII code of character "NUL"
- 0x23: ASCII code of character "#"

'GtSM0' (Disable Auto Trigger Mode)

Parameters: None

Format: GtSM0

Output: '@'

Purpose: Disable auto trigger mode.

[Example] Send 5 bytes (0x47, 0x74, 0x53, 0x4D, 0x30) to disable auto trigger mode.

- 0x47: ASCII code of character "G"
- 0x74: ASCII code of character "t"
- 0x53: ASCII code of character "S"
- 0x4D: ASCII code of character "M"
- 0x30: ASCII code of character "0"

'GtSM1' (Enable Auto Trigger Mode)

Parameters: None

Format: GtSM1

Output: '@'

Purpose: Disable auto trigger mode.

[Example] Send 5 bytes (0x47, 0x74, 0x53, 0x4D, 0x31) to enable auto trigger mode.

- 0x47: ASCII code of character "G"
- 0x74: ASCII code of character "t"
- 0x53: ASCII code of character "S"
- 0x4D: ASCII code of character "M"
- 0x31: ASCII code of character "1"

'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 KDC400 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 KDC400 to exit menu mode. KDC400 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

'GY3' (Display memory status)

Parameters: None

Format: 'GY3'

Output: '@'

Purpose: Display barcode data and the number of stored barcode after scanning a barcode

'GY5' (Display Barcode only)

Parameters: None

Format: 'GY5'

Output: '@'

Purpose: Display barcode data only

'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. KDC400 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,
 - KDC400 expects HOST to send either success or failure response to KDC400 once HOST receives packet data from KDC400.
 - 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 KDC400, HOST should send a failure response to KDC400 by sending a character '!'.
 - KDC400 tries multiple transmission up to 10 times if KDC400 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>

Value	Hex No.
0	0x00
1	0x01

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 48 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
32~47	"2", "0" ~ "2", "F"	32,30 ~ 32,46
48	"3", "0"	33,30

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>

Minimum Length	Hex. Number
2~48	0x02~0x30

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 KDC400

[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>

Value	Hex No.
0 ~ 10240	0x00 ~ 0x2800

Purpose: To get the number of stored barcodes in KDC400 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) - KDC410/415 Model

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

Format: 'O<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 3 bytes data (0x4F, 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#

[Example2] Send 10 bytes (0x4F, 0x46, 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
- OFFFFFFFFF#

[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' (Set barcode option) - KDC420/425 Model

- Parameters:** The options (it is unsigned two 32-bit numbers)
- Format:** 'O<1st option value>#<2nd 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 1st and 2nd option values.

1st option values

Barcodes	bit#	Options
Codabar	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
Reserved	5	Not used
	6	Not used
	7	Not used

Code 39	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
Reserved	13	Not used
	14	Not used
	15	Not used
I2of5	16	00: Do not verify check char.
	17	01: Verify check digit but do not transmit 10: Verify check digit and transmit
Reserved	18	Not used
	19	Not used
Code 11	20	0: Verify check digit 1: Verify check digits
Code128	21	0: ISBT concatenation off 1: ISBT concatenation on
Telepen	22	0: AIM output 1: Original output
Reserved	23	Not used
	24	Not used
	25	Not used
	26	Not used
	27	Not used
	28	Not used
	29	Not used
PosiCode	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
UPCA	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
UPCE	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
EAN-13	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
Postnet	20	0: Check digit do not transmit 1: Check digit transmit
PlanetCode	21	0: Check digit do not transmit 1: Check digit transmit
EAN-8	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
MSI	27	0: Verify check character, but do not transmit 1: Verify check character and transmit
EANUCC	28	0: Version off 1: Version on
	29	00: Emulation off
	30	01: 128 emulation 10: RSS emulation
Reserved	31	Not used

'o' (Read barcode option) - KDC410/415 Model

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 KDC400. Please see the table of command 'O' for the detailed explanation of options

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

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

'o' (Read barcode option) - KDC420/425 Model

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 KDC400. 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) - KDC410/415 Model

Parameters: None.

Format: 'P'

Output: A multi-byte string (see below)

Purpose: Get all barcodes data stored in KDC400 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)

- Bit 0 ~4 : Barcode type

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		

- Bit 5 : 1 if matching slave data, 0 if mismatching slave data

- Bit 6 : 1 if master data, 0 if slave data

- Bit 7 : 1 if data is collected in Master/Slave Application, 0 if data is collected in Normal mode

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] Output: 0x00, 0x00, 0x15, 0x11, 0x02, 0x30, 0x31, 0x32, 0x35, 0x34, 0x36, 0x36, 0x31, 0x32, 0x32, 0x39, 0x36, 0x00, 0x42, 0x17, 0x24

- N: 0x00, 0x00, 0x15 : 21 bytes
- C0: 0x11 : 17 bytes
- Y0: 0x02 : 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 all stored barcode) - KDC420/425 Model

Parameters: None.

Format: 'P'

Output: A multi-byte string (see below)

Purpose: Get all barcodes data stored in KDC400 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(2 bytes) including C/Y/D/T bytes

Y₀,Y₁,..., Y_n : 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 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] 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 nth stored barcode) - KDC410/415 Model

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 KDC400 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 ~4 : Barcode type

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		

- Bit 5 : 1 if matching slave data, 0 if mismatching slave data

- Bit 6 : 1 if master data, 0 if slave data

- Bit 7 : 1 if data is collected in master-slave mode, 0 if data is collected in normal mode

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)
- KDC400 tries multiple transmission up to 10 times if Handshake mode is enabled and KDC400 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 20th 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#

'p' (Download nth stored barcode) - KDC420/425 Model

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 KDC400 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)
- KDC400 tries multiple transmission up to 10 times if Handshake mode is enabled and KDC400 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 20th 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) - KDC410/415 Model

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

Format: 'S<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 option bits.
- Set the appropriate bit to “1” to enable an option

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		

Output: '!' – invalid parameter

'@' – all other times

Purpose: To enable/Disable symbologies

[Example1] Send 3 bytes data (0x53, 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#

[Example2] Send 10 bytes (0x53, 0x46, 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#

'S' (Select symbology) - KDC420/425 Model

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 11
	2	Code 32
	3	Code 39
	4	Code 93
	5	Code 128
	6	EAN-8
	7	EAN-13
	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
Reserved	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
2D symbologies	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
Reserved	10	Not used
	11	Not used
Postal codes	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
Reserved	21	Not used
	22	Not used
	23	Not used
OCR	24	OCR Off
	25	OCR-A
	26	OCR-B
	27	OCR U.S. Currency
	28	OCR MICR E-13B
	29	OCR Semi Font
Reserved	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, 0x46, 0x23, 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) - KDC410/415 Model

Parameters: None

Format: 's'

Output: '<value>@', where <value> is 4 bytes and @ means the end of <value>

- <value> should be interpreted according to the following table. Option is selected if corresponding bit is "1". Option is NOT selected if corresponding bit is "0".

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		

Purpose: To get enabled/disabled symbology options

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

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

's' (Read selected symbology) - KDC420/425 Model

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: KDC400 doesn't store scanned barcode in the memory and just transmits it to the host
- 1: KDC400 stores scanned data in the memory and transmits it to the host
- 2: KDC400 stores scanned data in the memory but doesn't transmit it to the host
- 3: KDC400 stores scanned data in the memory only if it succeeds to send read barcode to the host
- 4: KDC400 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 KDC400

'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: KDC400 doesn't store scanned barcode in the memory and just transmits it to the host
- 1: KDC400 stores scanned data in the memory and transmits it to the host
- 2: KDC400 stores scanned data in the memory but doesn't transmit it to the host
- 3: KDC400 stores scanned data in the memory only if it succeeds to send read barcode to the host
- 4: KDC400 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 KDC400 firmware version and @ means the end of data

Purpose: To read the firmware version of KDC400

'W' (Wake up)

Parameters: None

Format: "W"

Output: '@'

Purpose: To wake up KDC400 from sleep mode

[Note] KDC400 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: KDC400 sends barcode only to HOST,
- 1: KDC400 sends PACKET data and wait for response('@') from HOST

Output: '!' – invalid parameter

'@' – all other times

Purpose: KDC400 transmits scanned data to HOST if Wedge mode is enabled. User can set Wedge mode using 'U' command. KDC400 can send barcode data only or PACKET data

[Note]

- KDC400 tries multiple transmission up to 10 times if Handshake mode is enabled and KDC400 doesn't get response from HOST within 1 second
- KDC400 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) including N/C/Y/D/T/K but excluding S/@

C : Total number of barcode data(2 byte) including C/Y/D/T but excluding S/N/K/@

Y : Barcode type(1 byte)

D : Barcode data(variable size)

T : Timestamp(4 bytes)

K : Check sum(1 byte). $(N+C+Y+D+T+K) \bmod 256 = 0$

@: Termination character

'Z' (Set security level)

Parameters: Security level

Format: "Z<value>#", <value> is between 1 and 4, and # means the end of <value>

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.

Value	Hex No.
0	0x00
1	0x01
2	0x02
3	0x03

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

KDC400 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

- KDC400 does not perform handshaking if Wedging data format is packet data but handshake mode is disabled.

Packet data format and enabled Handshake mode

- KDC400 expects the HOST to send either success or failure response to KDC400 once HOST receives packet data from KDC400 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 KDC400, HOST should send a failure response to KDC400 by sending a character '!'.
- KDC400 tries multiple transmission up to 10 times if Handshake mode is enabled and KDC400 doesn't get response from HOST within 1 second

Miscellaneous

KDC400 sends "?" character

"?" is sent by KDC400 when the device receives undefined character (command) from Host.
Application ignores "?" character

KDC400 sends one to three leading "NULL" bytes

KDC400 *Bluetooth* module goes to sleep mode if not used for a while. KDC400 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

KOAMTAC 

CORPORATE HEADQUARTERS

116 Village Boulevard, Suite 200

Princeton, NJ 08540, USA

PH: +1-609-734-4335 FAX: +1-609-228-4373

email: info@koamtac.com

For more information, visit our website - www.koamtac.com