

REVISION

2.82

KoamTac
A New Wave in Auto ID

Programming Manual

KDC100



July, 2009

Copyright, License and Warning page

Copyright© 2009 by KoamTac, Inc. All rights reserved

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

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

KoamTac reserves the right to make changes to any product to improve reliability, function, or design. KoamTac doesn't assume any product liability arising out of, or in connection with, the application or use of any product, circuit, or application described herein.

Follow all warnings and instructions marked on manual and units

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

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

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



CAUTION

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

WARNING

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

INFORMATION TO USER:

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

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

CAUTION

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

DISPOSE USED BATTERIES ACCORDING TO THE INSTRUCTIONS

Table of Contents

Internal data buffer	4
<i>Data Format</i>	4
<i>How many barcodes can be stored?</i>	6
<i>What happens when the buffer is full?</i>	6
Commands Set	7
' <i>B</i> ' (<i>Read battery level</i>).....	9
' <i>C</i> ' (<i>Set date and time</i>).....	10
' <i>c</i> ' (<i>Read date and time</i>).....	11
' <i>D</i> ' (<i>Software trigger</i>).....	12
' <i>E</i> ' (<i>Erase all stored data</i>).....	13
' <i>e</i> ' (<i>Erase the last stored data</i>).....	14
' <i>F</i> ' (<i>Factory default</i>).....	15
' <i>GB0</i> ' (<i>Unlock button</i>).....	16
' <i>GB1</i> ' (<i>Lock button</i>).....	17
' <i>Gb0</i> ' (<i>Beeper off</i>).....	18
' <i>Gb1</i> ' (<i>Beeper on</i>).....	19
' <i>GC0</i> ' (<i>Disable port connection status display</i>).....	20
' <i>GC1</i> ' (<i>Enable port connection status display</i>).....	21
' <i>GD0</i> ' (<i>Unlock date and time setting</i>).....	22
' <i>GD1</i> ' (<i>Lock date and time setting</i>).....	23
' <i>GM</i> ' (<i>Message start position</i>).....	24
' <i>GMB0</i> ' (<i>Failure Alert Beep</i>).....	25
' <i>GMB1</i> ' (<i>Success Alert Beep</i>).....	26
' <i>GMS</i> ' (<i>Message display duration</i>).....	27
' <i>GMT</i> ' (<i>Display message</i>).....	28
' <i>GS0</i> ' (<i>Finish Synchronization</i>).....	29
' <i>GS1</i> ' (<i>Start Synchronization</i>).....	30
' <i>GTG</i> ' (<i>Read termination character</i>).....	32
' <i>GW</i> ' (<i>Menu Password Protection</i>).....	33
' <i>GX0</i> ' (<i>Disable Auto Menu Exit</i>).....	34
' <i>GX1</i> ' (<i>Enable Auto Menu Exit</i>).....	35
' <i>GY0</i> ' (<i>Display data, type, date and time</i>).....	36
' <i>GY1</i> ' (<i>Display data, date and time, battery status</i>).....	37
' <i>GY2</i> ' (<i>Display data, type, battery status</i>).....	38
' <i>H</i> ' (<i>Set handshake mode</i>).....	39
' <i>h</i> ' (<i>Read handshake mode</i>).....	40
' <i>L</i> ' (<i>Set minimum barcode length</i>).....	41

'l' (Read minimum barcode length).....	42
'M' (Read serial number).....	43
'N' (Get # of stored barcode).....	44
'O' (Set barcode option).....	45
'o' (Read barcode option).....	46
'P' (Download all stored barcode).....	47
'p' (Download n th stored barcode).....	49
'S' (Select symbology).....	51
's' (Read selected symbology).....	53
'T' (Set Timeout).....	54
't' (Read timeout value).....	55
'U' (Set data process mode).....	56
'u' (Read data process mode).....	57
'V' (Read firmware version).....	58
'W' (Wake up).....	59
'w' (Set data format).....	60
'Z' (Set security level).....	61
'z' (Read security level).....	62
Wedging Data Format and Handshake Mode	63
Barcode only format.....	63
Packet data format and disabled Handshake mode.....	63
Packet data format and enabled Handshake mode.....	63
Miscellaneous	64
KDC100 sends "?" character.....	64
Contact Information	65

KoamTac KDC100 provides easy and extensive commands set to application developers who wish to control the KDC100 and resulting data for their application.

Internal data buffer

KDC100 has 190K 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 190K 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

KDC100 stores the read barcodes in the internal flash memory in the following data format:

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

Where

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

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

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 barcode plus 1 (1 byte), then the barcode type (1 byte), and then the barcode value (without the string terminator '\0').

The timestamp field has 6 sub-fields as follows:

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

Note:

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

User Applications generated by Application Generator

KDC100 stores the read barcodes in the internal flash memory in the following data format:

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

Where

- C0, C1, ..., Cn : Total number of each barcode data (1 byte)
 - **2 bytes for KDC200P/KDC300, 1 byte for KDC100/200**
- Y0, Y1, ..., Yn : Type of each barcode(1 byte)
 - 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
 - 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)
 - The 1st byte indicates Predefined or User Application
 - 0xFF : User Application
 - 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 barcode plus 1 (1 byte), then the barcode type (1 byte), and then the barcode value (without the string terminator '\0').

The timestamp field has 6 sub-fields as follows:

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

Note:

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

How many barcodes can be stored?

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

Example: If only UPCA barcodes are scanned and the check digit is not transmitted, then each barcode takes up 11 (barcode data) + 2 (length and type) + 4 (time stamp) = 17 bytes. The maximum number of UPCA barcodes that can be saved is $204,800/17 = 12,047$. However, the maximum number of barcodes can be stored is 10,240. Therefore, KDC100 can store 10,240 UPCA barcodes.

What happens when the buffer is full?

When the internal flash memory is full, KDC100 displays “Buffer Full” message and ignores the command to scan barcodes.

Commands Set

KDC100 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 KDC100 before sending individual commands.

Command	Input	Output	Description	
B	None	<i>value</i> @	Returns current battery capacity(<i>value</i> = 0~100)	
C	YMDHmS	@	Setting new date and time of KDC100	
c	None	YMDHmS @	Returns current date and time of KDC100	
D	None	@ or !	Reading barcode (Software trigger)	
E	None	@	Erase all stored barcodes	
e	None	@	Erase the last stored barcode	
F	None	@	Set internal options to factory default setting	
G	B0	@	Unlock buttons for normal operation	
	B1	@	Lock buttons to prevent undesired accidental operation	
	b0	@	Disable beep sound	
	b1	@	Enable beep sound	
	C0	@	Disable Serial and USB connection status display	
	C1	@	Enable Serial and USB connection status display	
	D0	@	Unlock date and time setting	
	D1	@	Lock date and time setting	
	M	< <i>value</i> # ; <i>value</i> #>#	@ or !	Set start display position of message from Host
		S< <i>value</i> >#	@ or !	Set message display duration from Host
		T< <i>value</i> ><CR>	@	Display message from Host on KDC
		MB0	@	Failure Alert Beep
		MB1	@	Success Alert Beep
	S0	@	Start Synchronization Mode	
	S1	@	Finish Synchronization Mode	
	TS	<i>Value</i> #	@ or !	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< <i>password</i> >#	@	Enable menu password protection
	X	0	@	Disable automatic menu exit feature
		1	@	Enable automatic menu exit feature
	Y0	None	@	Display scanned barcode data, barcode type, date and time
	Y1	None	@	Display scanned barcode data, date and time,

				battery status
	Y0	Note	@	Display scanned barcode data, barcode type, battery status
H	<i>value #</i>		@	Enable/Disable handshake mode while transferring packet data
h	None		<i>value @</i>	Get handshake state
L	<i>value #</i>		@ or !	Set minimum barcode length
l	None		<i>value @</i>	Get minimum barcode length
M	None		<i>value @</i>	Get serial number of KDC
N	None		<i>value @</i>	Get number of barcodes stored
O	<i>value #</i>		@ or !	Set scan options and flags
o	None		<i>value @</i>	Get scan options and flags
P	None		<i>Data</i>	Upload all stored barcode data
p	<i>value #</i>		<i>Data</i>	Upload N _{th} stored barcode
S	<i>value #</i>		@ or !	Set decoding symbologies
s	None		<i>value @</i>	Get current decoding symbologies
T	<i>value #</i>		@ or !	Set decoding timeout
t	None		<i>value @</i>	Get current barcode read timeout
U	<i>value #</i>		@ or !	Set data process mode (Wedge, Store)
u	None		<i>value @</i>	Get barcode handling mode selected by command 'U'
V	None		<i>string @</i>	Get firmware version
W	None		@	Wake up KDC100 from sleep mode
w	<i>value #</i>		@ or !	Select wedging data format (Barcode only, packet data)
Z	<i>value #</i>		@ or !	Set security level
z	None		<i>value @</i>	Get current security level

'B' (Read battery level)

Parameters: None

Format: 'B'

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

Purpose: Read the current battery capacity of KDC100

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

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

'C' (Set date and time)

Parameters: Date and Time value to be set in KDC100

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 KDC100

[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 KDC100

[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: '!' – KDC100 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 all stored data)

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 KDC100 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 KDC100 up, down and scan buttons for normal operation

'GB1' (Lock button)

Parameters: None

Format: 'GB1'

Output: '@'

Purpose: To lock KDC100 up, down and scan buttons. Locking buttons prevents undesired accidental operation and keeps KDC100 in sleep mode (battery saving mode)

'Gb0' (Beeper off)

Parameters: None

Format: 'Gb0'

Output: '@'

Purpose: Disable KDC100 beep sound

'Gb1' (Beeper on)

Parameters: None

Format: 'Gb1'

Output: '@'

Purpose: To enable KDC100 beep sound

'GC0' (Disable port connection status display)

Parameters: None

Format: 'GC0'

Output: '@'

Purpose: Disable Serial and USB connection status display. KDC100 does not display connected and disconnected message.

'GC1' (Enable port connection status display)

Parameters: None

Format: 'GC1'

Output: '@'

Purpose: Enable Serial and USB connection status display. KDC100 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 KDC100 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 KDC100 menu.

'GM' (Message start position)

Parameters: row # and column #

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

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

Output: '!' – Invalid parameter

'@' – Set start position successfully

Purpose: To set start position of message

[Note] Default position is the 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

'GMS' (Message display duration)

Parameters: Message display duration in seconds

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

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

Output: '!' – Invalid parameter

'@' – Set start position successfully

Purpose: To set the message display duration

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

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

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

'GMT' (Display message)

Parameters: Character string to be displayed on KDC

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

Output: '@'

Purpose: Display message on KDC

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

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

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

'GSO' (Finish Synchronization)

Parameters: None

Format: 'GSO'

Output: '@'

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

'GS1' (Start Synchronization)

Parameters: None

Format: 'GS1'

Output: '@'

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

GTS' (Set termination character)

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

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

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

Output: '!' – Invalid parameter

'@' – Set termination character successfully

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

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

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

'GTG' (Read termination character)

Parameters: None

Format: 'GTG'

Output: '<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

'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 KDC100 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 KDC100 to exit menu mode. KDC100 automatically exits from menu mode if it remains idle in menu mode for 5 minutes.

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

Parameters: None

Format: 'GY0'

Output: '@'

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

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

Parameters: None

Format: 'GY1'

Output: '@'

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

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

Parameters: None

Format: 'GY2'

Output: '@'

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

'H' (Set handshake mode)

Parameters: 0 or 1 in hexadecimal

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

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

Output: '!' – invalid parameter

'@' – all other times

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

[Note]

- Handshake mode is applicable for only 'p' command and wedging in packet data format
- Handshake mode is NOT applicable for 'P' command and wedging in data only format
- If handshake mode is enabled,
 - KDC100 expects HOST to send either success or failure response to KDC100 once HOST receives packet data from KDC100.
 - 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 KDC100, HOST should send a failure response to KDC100 by sending a character '!'.
 - KDC100 tries multiple transmission up to 10 times if KDC100 doesn't get response within 1 second or receives failure response '!' from HOST.

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

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

'h' (Read handshake mode)

Parameters: None

Format: 'h'

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

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

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

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

'L' (Set minimum barcode length)

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

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

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

Output: '!' – invalid parameter

'@' – all other times

Purpose: Set the minimum length of barcodes

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

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

'I' (Read minimum barcode length)

Parameters: None

Format: 'I'

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

Purpose: To get the current setting of minimum barcode length

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

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

'M' (Read serial number)

Parameters: None

Format: 'M'

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

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

Purpose: To get the serial number of KDC100

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

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

'N' (Get # of stored barcode)

Parameters: None

Format: 'N'

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

Purpose: To get the number of stored barcodes in KDC100 internal flash memory

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

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

'O' (Set barcode option)

Parameters: The options (it is unsigned 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
- O0#

[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' (Read barcode option)

Parameters: None

Format: 'o'

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

Purpose: To get the barcode options saved in KDC100. 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

'P' (Download all stored barcode)

Parameters: None.

Format: 'P'

Output: A multi-byte string (see below)

Purpose: Get all barcodes data stored in KDC100 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 nth stored barcode)

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

Format: 'p<value>#' where <value> is the number of barcode data stored in the internal flash to upload in hexadecimal and # means the end of <value>. The first stored barcode location is "0", not "1"

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

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

Purpose: Get the Nth barcode data stored in KDC100 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 Application, 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)
- KDC100 tries multiple transmission up to 10 times if Handshake mode is enabled and KDC100 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)

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' (Read selected symbology)

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

'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: KDC100 doesn't store scanned barcode in the memory and just transmits it to the host
- 1: KDC100 stores scanned data in the memory and transmits it to the host
- 2: KDC100 stores scanned data in the memory but doesn't transmit it to the host
- 3: KDC100 stores scanned data in the memory only if it succeeds to send read barcode to the host
- 4: KDC100 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 KDC100

'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: KDC100 doesn't store scanned barcode in the memory and just transmits it to the host
- 1: KDC100 stores scanned data in the memory and transmits it to the host
- 2: KDC100 stores scanned data in the memory but doesn't transmit it to the host
- 3: KDC100 stores scanned data in the memory only if it succeeds to send read barcode to the host
- 4: KDC100 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 KDC100 firmware version and @ means the end of data

Purpose: To read the firmware version of KDC100

'W' (Wake up)

Parameters: None

Format: "W"

Output: '@'

Purpose: To wake up KDC100 from sleep mode

[Note] KDC100 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: KDC100 sends barcode only to HOST,
- 1: KDC100 sends PACKET data and wait for response('@') from HOST

Output: '!' – invalid parameter

'@' – all other times

Purpose: KDC100 transmits scanned data to HOST if Wedge mode is enabled. User can set Wedge mode using 'U' command. KDC100 can send barcode data only or PACKET data

[Note]

- KDC100 tries multiple transmission up to 10 times if Handshake mode is enabled and KDC100 doesn't get response from HOST within 1 second
- KDC100 supports two kind of data format when sending data to HOST

– Barcode only

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

where: Barcode data is the actual read barcode data and termination characters are "None", "CR (0x0d, '\r')", "LF (0x0a, '\n') or "CR + LF"

– PACKET DATA

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

where:

S : Start byte 0x03(1 byte)

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

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

Y : Type of each barcode(1 byte)

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

T : Timestamp of each barcode(4 bytes)

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

'Z' (Set security level)

Parameters: Security level

Format: "Z<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.

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

KDC100 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

- KDC100 does not perform handshaking if Wedging data format is packet data but handshake mode is disabled.

Packet data format and enabled Handshake mode

- KDC100 expects the HOST to send either success or failure response to KDC100 once HOST receives packet data from KDC100 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 KDC100, HOST should send a failure response to KDC100 by sending a character '!'.
- KDC100 tries multiple transmission up to 10 times if Handshake mode is enabled and KDC100 doesn't get response from HOST within 1 second

Miscellaneous

KDC100 sends "?" character

"?" is sent by KDC100 when the device receives undefined character (command) from Host.
Application ignores "?" character

Contact Information



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

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