

Barcode Printer JavaScript SDK User Manual (V2.0)



Barcode Printer JavaScript SDK preparation

1

1. Install USBDK :

For Windows : open <sample code folder>\Server\libusb\windows,

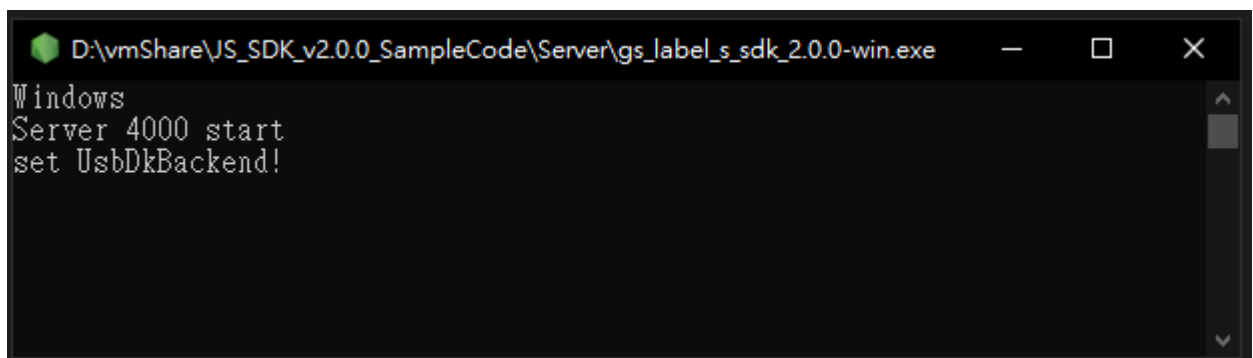
and double-click UsbDk_1.0.22_x64.msi to install USBDK .

For Linux : open <sample code folder>/Server/libusb/linux .

type the command : `#bash linux_libusb_init.sh` and follow the instructions to input user password. After setting, please re-plug USB device . Just have to prepare one time .

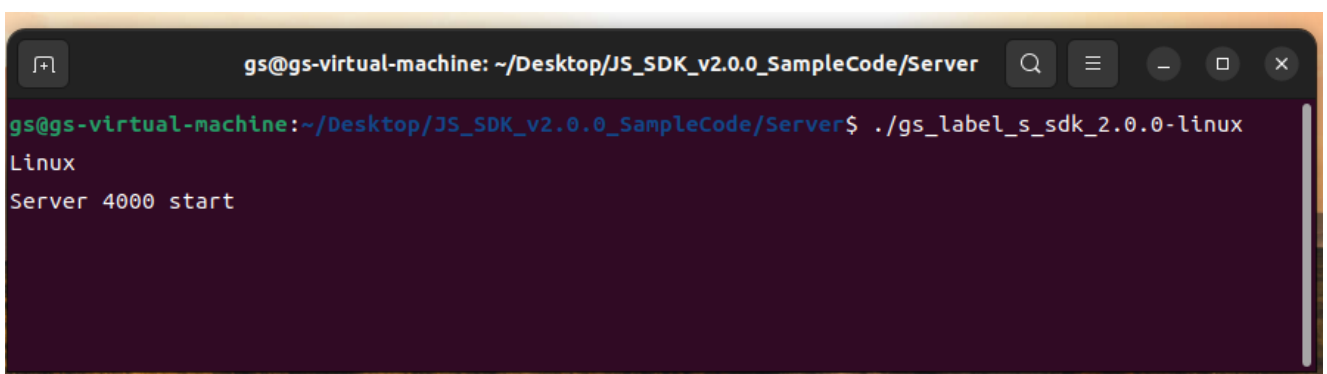
2. Start Printer Server console: In server folder(<sample code folder>/Server) .

For Windows, double-click `gs_label_s_sdk_2.0.0-win.exe` .



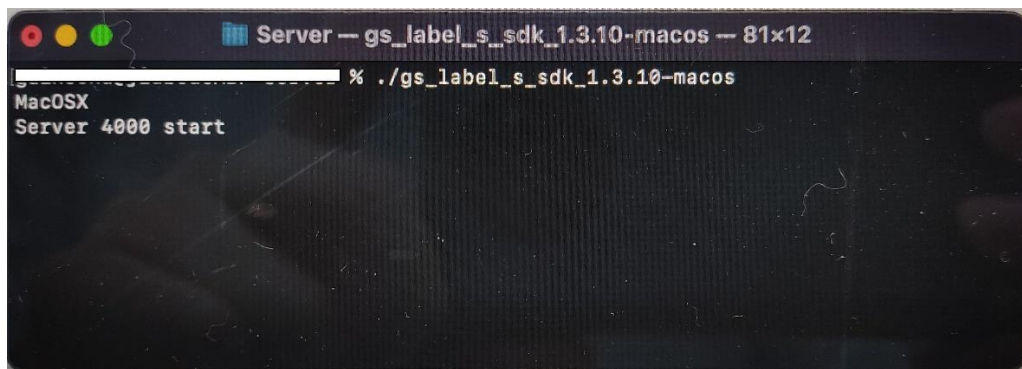
```
D:\vmShare\JS_SDK_v2.0.0_SampleCode\Server\gs_label_s_sdk_2.0.0-win.exe
Windows
Server 4000 start
set UsbDkBackend!
```

For Linux , type the command : `#./gs_label_s_sdk_2.0.0-linux` .



```
gs@gs-virtual-machine: ~/Desktop/JS_SDK_v2.0.0_SampleCode/Server
gs@gs-virtual-machine:~/Desktop/JS_SDK_v2.0.0_SampleCode/Server$ ./gs_label_s_sdk_2.0.0-linux
Linux
Server 4000 start
```

for Mac · type the command : `./gs_label_s_sdk_2.0.0-macos`



After above action, the printer server console will start · and server run at 4000 port ·

3. Add `gs_label_c_sdk_2.0.0.js` into web html file with `<script>` :

```
<script type="text/javascript" src="gs_label_c_sdk_2.0.0.js"> </script>
```

Version 2.0 has made structural adjustments, requiring the use of corresponding .js files and server-side tools.

Barcode Printer JavaScript SDK User Manual (V2.0)

1. Create GSDevice object

```
var devInterface = " ethernet"
var printerIP=" 192.168.1.128" (barcode print IP address)
var printerPort=9100
var serverIP=" 192.168.1" ( server IP address, pc connect to Printer )
var gsobj= new gs.GSDevice(printerIP, printerPort, serverIP, devInterface);
```

2. GSDevice Object API

constructor		
Description	Initial GSDevice object	
Syntax	GSDevice(printerIP, port, serverIP, devInterface)	
Parameter	printerIP	Specify Printer IP address
	port	Specify Printer Port
	server IP	Specify Server IP (server at localhost · fill in 127.0.0.1)
	devInterface	Select communicate Interface (when select USB · ignore printerIP and Port args)
connect method		
Description	Connect to printer server console and then connect to barcode printer	

Syntax	connect(callback)	
Parameter	callback	callback function for message of connection (OK/Disconnected)

disconnect method		
Description	Disconnect to printer server console and then disconnect to barcode printer	
Syntax	disconnect(callback)	
Parameter	callback	callback function for message of disconnection

3. Printer API

setup Method		
Description	Setup label width, label height, print speed, density, sensor type, gap/blackmark vertical distance and gap/blackmark offset distance.	
Syntax	setup(width, height, speed, density, label type, distance, offset)	
Parameter	Type	Description
width	float	Set up label width · unit : mm
height	float	Set up label height · unit: mm

speed	int	Defines the print speed (1~15: print speed at 1"~15"/sec) Selectable print speeds depend on different printer models, and maximum speed is 15"/sec.
density	Int	Set up print density (0~15); If the number is bigger, the printout will be darker.
label type	int	Set up label sensor type ; 0 : gap sensor 1 : black mark sensor
distance	float	Set up vertical gap height of the gap/black mark ° Unit: mm
offset	float	Set up offset distance of the gap/black mark, unit: mm, this parameter is set to 0 when the general label is used.
setDirectionAndMirror method		
Description	This method defines the printout direction and mirror image.	
Syntax	setDirectionAndMirror(direction,mirror)	
Parameter	Type	Description

direction	int	0: Top out 1: Bottom out
mirror	int	0: Print normal image 1: Print mirror image
setShift method		
Description	<p>This method moves the label's vertical position.</p> <p>A positive value will cause the image to move towards the printing direction, and a negative value will cause the image to move away from the printing direction .</p>	
Syntax	setShift(shiftY)	
Parameter	Type	Description
shiftY	int	<p>The maximum value is 1 inch.</p> <p>For 200 dpi printers, the range is –203 to 203; for 300 dpi printers, the range is –300 to 300.</p> <p>The unit is dot.</p>
printReverse method		
Description	This method reverses a region in image buffer	
Syntax	printReverse(x_start, y_start, x_width, y_height)	
Parameter	Type	Description

x_start	int	The x-coordinate of the starting point (in dots)
y_start	int	The Y-coordinate of the starting point (in dots)
x_width	int	X-axis region width (in dots)
y_height	int	Y-axis region width (in dots)
setOffset method		
Description	<p>This method defines the selective, extra label feeding length each form feed takes, which, especially in peel-off mode and cutter mode, is used to adjust label stop position, so as for label to register at proper places for the intended purposes.</p> <p>The printer back tracks the extra feeding length before the next run of printing.</p>	
Syntax	setOffset(offset)	
Parameter	Type	Description
offset	float	<p>The offset distance(mm)</p> <p>$1 \leq m \leq 1$ (inch)</p>
setCutMode method		
Description	This method defines cutter mode and piece	
Syntax	setCutMode(mode, piece)	
Parameter	Type	Description

mode	int	cutter mode · default: 1 0 : using Partial Cutter 1 : using Cutter
piece	int	Set number of printing labels per cut. (0~65535)
setAfterPrintAction method		
Description	Set Action after print.	
Syntax	setAfterPrintAction(mode)	
Parameter	Type	Description
mode	int	set Action · Default: 1 0 : Stay 1 : Tear 2 : Peel 3 : Cut
genericDefault method		
Description	To reset printer settings to default.	
Syntax	genericDefault()	
sensorDefault method		
Description	To reset sensor settings to default.	
Syntax	sensorDefault ()	

rfidSetupDefault method

Description To reset RFID settings to default.

Syntax rfidSetupDefault ()

clearbuffer method

Description Clear the image buffer.

Syntax clearbuffer ()

formfeed method

Description Feed label to the top of next label.
This function should be used after setup function.

Syntax formfeed ()

nobackfeed method

Description Set the paper not to back feed.

Syntax nobackfeed ()

barcode method

Description Use built-in barcode formats to print barcodes.

Syntax barcode(x, y, type, height, readable, rotation, narrow, wide, content)

Parameter	Type	Description
x	int	Specify the x-coordinate bar code on the label,

		Unit: dot
y	int	Specify the y-coordinate bar code on the label, Unit: dot
type	String	Set up Code Type · refer to Appendix
height	int	Set up bar code height (in dots)
readable	int	Set up whether to print human recognizable interpretation (text) or not. 0: Do not print 1: Print barcode document left 2: Print barcode code document in the center 3: Print barcode document right
rotation	int	Set up barcode rotation 0 : No rotation 90 : Rotate 90 degrees clockwise 180 : Rotate 180 degrees clockwise 270 : Rotate 270 degrees clockwise
narrow	int	Set up narrow bar ratio (in dots), refer to Appendix
wide	int	Set up wide bar ratio (in dots), refer to Appendix
content	String	Content of barcode.

		Please note that the maximum number of digits of bar code content. refer to Appendix
QRcode method		
Description	Use built-in QR code formats to print QR code	
Syntax	printQRcode(x, y, eccLevel, cellWidth, mode, rotation, content,{ justification, model, mask, area })	
Parameter	Type	Description
x	int	The upper left corner x-coordinate of the QR code
y	int	The upper left corner y-coordinate of the QR code
eccLevel	String	Error correction recovery level L : 7% 、 M : 15% 、 Q : 25% 、 H : 30%
cellWidth	int	QR code shape width 1~10
mode	String	Set QR code mode : A : Auto 、 M : Manual
rotation	int	Set up QR code rotation : 0 : 0 degree 、 90 : 90 degree 、 180 : 180 degree 、 270 : 270 degree
content	String	The encodable character set is described as below,

		<p>*Encodable character set:</p> <ol style="list-style-type: none"> 1) Numeric data: (digits 0~9) 2) Alphanumeric data <ul style="list-style-type: none"> Digits 0-9 Upper case letters A-Z Nine other characters: space, \$ % * + - . / :) 3) 8-bit byte data <ul style="list-style-type: none"> JIS 8-bit character set (Latin and Kana) in accordance with JIS X 0201 4) Kanji characters <ul style="list-style-type: none"> Shift JIS values 8140HEX –9FFCHEX and E040HEX –EAA4 HEX. These are values shifted from those of JIS X 0208. Refer to JIS X 0208 Annex 1 <p>Shift Coded Representation for detail.</p>
justification	String	<p>(optional)Barcode justification (J1 to J9 valid , default : J1)</p> <p>J1:top-left; J2:top-middle; J3:top-right;</p>

		J4:middle-left; J5:center; J6: middle-right; J7:bottom-left; J8:bottom-middle; J9:bottom-right;
model	String	(optional)Set up original /enhanced QR code M1: (default), original version M2: enhanced version (Almost smart phone is supported by this version.)
mask	String	(optional)Mask code S0~S8 · default : S7
area	String	(optional)Maximum size of barcode area (Xdots; ex: X100);
printfont method		
Description	Use printer built-in fonts to print	
Syntax	printfont(x, y, fontname, rotation, x_scale, y_scale, text)	
Parameter	Type	Description
x	int	The x-coordinate of the text
y	int	The y-coordinate of the text
fontname	String	Built-in font type 1: 8*/12 dots 2: 12*20 dots

		3: 16*24 dots 4: 24*32 dots 5: 32*48 dots TST24.BF2: Traditional Chinese 24*24 TST16.BF2: Traditional Chinese 16*16 TSS24.BF2: Simplified Chinese 24*24 TSS16.BF2: Simplified Chinese 16*16
rotation	int	The rotation angle of text 0 : No rotation 90: degrees, in clockwise direction 180 : degrees, in clockwise direction 270 : degrees, in clockwise direction
x_scale	int	Horizontal multiplication, Available factors: 1~10
y_scale	int	Vertical multiplication, Available factors: 1~10
text	String	Content of text string
printblock method		
Description	Use printer built-in fonts to print paragraph.	
Syntax	printblock(x, y, width, height, fontname, rotation, xscale, yscale, space ,align ,content)	

Parameter	Type	Description
x	int	The x-coordinate of the text (in dots)
y	int	The y-coordinate of the text (in dots)
width	Int	The width of block for the paragraph in dots
height	int	The height of block for the paragraph in dots
fontname	String	<p>Built-in font type</p> <p>1: 8*12 dots</p> <p>2: 12*20 dots</p> <p>3: 16*24 dots</p> <p>4: 24*32 dots</p> <p>5: 32*48 dots</p> <p>TST24.BF2: Traditional Chinese 24*24</p> <p>TST16.BF2: Traditional Chinese 16*16</p> <p>TSS24.BF2: Simplified Chinese 24*24</p> <p>TSS16.BF2: Simplified Chinese 16*16</p>
rotation	int	<p>The rotation angle of text</p> <p>0 : No rotation</p> <p>90 : degrees, in clockwise direction</p> <p>180 : degrees, in clockwise direction</p>

		270 : degrees, in clockwise direction
x_scale	int	Horizontal multiplication, Available factors: 1~10
y_scale	int	Vertical multiplication, Available factors: 1~10
space	int	Add or delete the space between lines (in dots)
align	int	Text alignment 0 : default (Left) 1 : Left 2 : Center 3 : Right
content	String	Data in block. The maximum data length is 4092 bytes.
printlabel method		
Description	Print the label format currently stored in the image buffer.	
Syntax	printlabel(set, copy)	
Parameter	Type	Description
set	int	Specifies how many sets of labels will be printed. $1 \leq \text{set} \leq 999999999$
copy	int	Specifies how many copies should be printed for each particular label set. $1 \leq \text{copy} \leq 999999999$

downloadbmp method

Description	Download the monochrome BMP file to main board flash memory	
Syntax	downloadbmp(filedata, filename)	
Parameter	Type	Description
filedata	ArrayBuffer	BMP Binary data (1 bit color format)
filename	String	BMP file name.

downloadpcx method

Description	Download the monochrome PCX file to main board flash memory	
Syntax	downloadpcx(filedata, filename)	
Parameter	Type	Description
filedata	ArrayBuffer	PCX Binary data (1 bit color format)
filename	String	PCX file name

sendCommand method

Description	Sent built-in commands to the printer.	
Syntax	sendCommand(command)	
Parameter	Type	Description
command	String	Refer to TSPL programming manual commands

		for details.
printerstatus method		
Description	Obatin the printer status.	
Syntax	printerstatus (callback)	
Parameter	Type	Description
callback	function	Callback function for printer status message.
Return message	Normal	
	Head opened	
	Paper Jam	
	Paper Jam and head opened	
	Out of paper	
	Out of paper and head opened	
	Out of ribbon	
	Out of ribbon and head opened	
	Out of ribbon and paper jam	
	Out of ribbon, paper jam and head opened	
	Out of ribbon and out of paper	
	Out of ribbon, out of paper and head opened	

	Pause	
	Printing	
	Other error	
labelCalibration method		
Description	Auto calibration for RFID label	
Syntax	labelCalibration (mode, type, cb)	
Parameter	Type	Description
mode	string	A : Auto mode
type	string	Label Type ◦ R : RFID
cb	function	Callback function get send status("Sent!" or "Fail")
writeUHF method		
Description	Write data to UHF tag memory.	
Syntax	writeUHF(dataFormat, startBlockNo, byteSize, Gen2MemoryBank, datastring)	
Parameter	Type	Description
dataFormat	String	Define data format ◦ default is "H" A : ASCII H : Hexadecimal

startBlockNo	int	Secify the 16-bit starting block number Default : 2
byteSize	int	Values: 1 to n, where n is the maximum number of bytes for the tag. Default: 1
Gen2MemoryBank	String	Select Gen2 memory bank R : Reserved E : EPC (Default) T : TID(Tag ID) U : User
datastring	String	Data string.

EPCPWD_Action method

Description	Lock or unlock EPC memory with password for UHF GEN2 tag.	
Syntax	EPCPWD_Action (action,password)	
Parameter	Type	Description
action	String	Action type U : unlock EPC memory bank L : lock EPC memory bank O : permanently unlock EPC memory bank

		P : permanently lock EPC memory bank
password	String	password · 8 HEX characters. (0~9, A,B,C,D,E,F)
TIDPWD_Action method		
Description	Lock or unlock TID memory with password for UHF GEN2 tag.	
Syntax	TIDPWD_Action(action,password)	
Parameter	Type	Description
action	String	Action type U : unlock TID memory bank L : lock TID memory bank O : permanently unlock TID memory bank P : permanently lock TID memory bank
password	String	password · 8 HEX characters. (0~9, A,B,C,D,E,F)
USERPWD_Action method		
Description	Lock or unlock USER memory with password for UHF GEN2 tag.	
Syntax	USERPWD_Action(action,password)	
Parameter	Type	Description
action	String	Action type

		U : unlock USER memory bank L : lock USER memory bank O : permanently unlock USER memory bank P : permanently lock USER memory bank
password	String	password · 8 HEX characters. (0~9, A,B,C,D,E,F)
AccessPWD_Action method		
Description	Lock or unlock access password with password for UHF GEN2 tag.	
Syntax	AccessPWD_Action(action,password)	
Parameter	Type	Description
action	String	Action type U : unlock the access password* L : lock the access password* O : permanently unlock the access password P : permanently lock the access password S : Set Password
password	String	password · 8 HEX characters. (0~9, A,B,C,D,E,F)

KillPWD_Action method

Description	Lock or unlock kill password with password for UHF GEN2 tag.	
Syntax	KillPWD_Action(action,password)	
Parameter	Type	Description
action	String	Action type U : unlock the kill password* L : lock the kill password* O : permanently unlock the kill password P : permanently lock the kill password S : Set Password
password	String	password · 8 HEX characters. (0~9, A,B,C,D,E,F)

Set_RFIDProcedure method

Description	Set RFID procedure	
Syntax	Set_RFIDProcedure(tagType, rw_position, void_printout, tryEncodie_times, error_handle, speed,retry_times,[dpi])	
Parameter	Type	Description
tagType	int	Set Tag type · accepted value:1~10 · For UHF:

		<p>1 = ISO 18000 6C/Class 1 Gen2 (Q command)</p> <p>8 = ISO 18000 6C/Class 1 Gen 2 (default)</p> <p>For HF</p> <p>10 = UHF-J</p>
rw_position	int	<p>Move the media to the specified position on the label, measured in dot rows from the label top, before encoding</p> <p>Accept value: 0~9999(dot) · default is 0.</p> <ul style="list-style-type: none"> when using dpi parameter , rw_position unit as mm.
void_printout	int	<p>Set the length of the void printout in vertical (Y axis) dot rows.</p> <p>Accepted values: 0 to label length</p> <p>Default: label length</p> <ul style="list-style-type: none"> when using dpi parameter , rw_position unit as mm.
tryEncodie_times	int	<p>The number of labels that will be attempted in case of read/encode failure. Accepted</p>

		<p>values: 1 to 10</p> <p>Default: 3</p>
error_handle	string	<p>If an error persists after the specified number of labels are tried, perform this error handling action.</p> <p>N : No action (Default)</p> <p>P : Pause mode</p> <p>E : Error mode</p>
speed	int	<p>If a label is voided, the speed at which "VOID" will be printed across the label.</p> <p>Accepted value: 2~10(IPS),</p> <p>Default is 2.</p>
retry_times	int	<p>The retry times of a tag that will be attempted in case of read/encode failure.</p> <p>Accepted value:0~10 · Default is 6</p>
[dpi]	string	<p>Optional parameter ·</p> <p>when no dpi parameter or dpi is null, unit of rw_position and void_printou is dot ·</p> <p>when dpi is "203" or "300" ·</p>

		unit of rw_position and void_printou is mm °
writeHF method		
Description	Write data to HF tag memory.	
Syntax	writeHF (dataFormat, startBlockNo, byteSize, datastring)	
Parameter	Type	Description
dataFormat	String	Define data format · default is "H" A : ASCII H : Hexadecimal
startBlockNo	int	Secify the 16-bit starting block number · Default : 2
byteSize	int	Values: 1 to n, where n is the maximum number of bytes for the tag. Default: 1
datastring	String	Data string
readUHF method		
Description	Read data from UHF tag memory (R command)	
Syntax	readUHF(dataFormat, startBlockNo, byteSize, Gen2MemoryBank, callback)	
Parameter	Type	Description

callback	function	Callback function for read data of RFID . If Error occurred · return message with error code · refer to Appendix II
dataFormat	String	Setting callback returned data format . A : ASCII H : Hexadecimal (Default)
startBlockNo	int	Secify the 16-bit starting block number to read · Default is 0
byteSize	int	Secify the data lengths to read · default is 1
Gen2MemoryBank	String	Gen2 memory bank . R = Reserved E = EPC T = TID U = UESR Default : E
queryUHF method		
Description	Read data from UHF tag memory (Q command)	
Syntax	queryUHF(dataFormat, pcRetureSetting, crc16ReturnSetting, callback_epcNumber)	

Parameter	Type	Description
dataFormat	String	Set up callback return data format · A : ASCII H : Hexadecimal (Default)
pcReturnSetting	Int	enable/disable PC value returned 0 : read epc data not include PC value 1 : read epc data include PC value
crc16ReturnSetting	int	enable/disable CRC-16 value returned 0 : read epc data not include CRC-16 value 1 : read epc data include CRC-16 value
callback_epcNumber	function	Callback function for read data of RFID Ex1: query UHF not includes PC and CRC16 : UHF QUERY H,0,0 AAAABBBB Ex2: query UHF includes PC and CRC16 : UHF QUERY H,1,1 1000AABBCCDDEEFFC7AC

		If Error occurred · return message with error code · refer to Appendix II
writeGJB method		
Description	Write data to UHF GJB tag memory	
Syntax	writeGJB(dataFormat, startBlockNo, byteSize, GJBMemoryBank, datastring, writePassword)	
Parameter	Type	Description
dataFormat	String	Define data format · default is "H" A : ASCII H : Hexadecimal
startBlockNo	int	Secify the 16-bit starting block number · Default : 1
byteSize	int	Values: 1 to n, where n is the maximum number of bytes for the tag. Default: 1
GJBMemoryBank	String	Select GJB memory bank to write R = Reserved E = EPC T = TID

		U = User 2 = User 2 3 = User 3 Default : E
datastring	String	Data string
writePassword	String	Writing password · 8 HEX characters. (0~9, A,B,C,D,E,F)
readGJB method		
Description	Read data from UHF GJB tag.	
Syntax	readGJB(dataFormat, startBlockNo, byteSize, GJBMemoryBank, readPassword, callback_gjbddata)	
Parameter	Type	Description
dataFormat	String	Set up callback returned data format · A : ASCII H : Hexadecimal (Default)
startBlockNo	int	Secify the 16-bit starting block number to read · Default is 0
byteSize	int	Secify the data lengths to read · default is 1
GJBMemoryBank	String	Select GJB memory bank to read ·

		E = EPC T = TID U = User 2 = User 2 3 = User 3 Default : E
readPassword	String	Reading password · 8 HEX characters. (0~9, A,B,C,D,E,F)
callback_gjbddata	Function	Callback function for read data of GJB tag.
setState_Action method		
Description	Set memory status with password for UHF GJB tag.	
Syntax	setState_Action(GJBMemoryBank, action, statePwd)	
Parameter	Type	Description
GJBMemoryBank	String	Select GJB memory bank for set, E = EPC (default) T = TID U = User 2 = User 2 3 = User 3

action	String	<p>Memory Status type</p> <p>A : Lock0(readable and writable)</p> <p>B : Lock1(read only)</p> <p>C : Lock2(write only)</p> <p>D : Lock3(non-readable and non-writable)</p> <p>*Each Memory support status :</p> <ul style="list-style-type: none"> EPC area : A:read and write B: read only USER area(include User 、 User2 、 User3) : A:read and write B:read only C: write only D: not allow access TID area : B:read only D: not allow access SAFE area : C: write only D: not allow access
statePwd	String	<p>Status password ·</p> <p>8 HEX characters. (0~9, A,B,C,D,E,F)</p>
setPWD_Action method		
Description	Set Write/Read/Status/Kill Password to UHF GJB tag.	
Syntax	setPWD_Action(pwdArea, action, newPwd, writePwd)	
Parameter	Type	Description
pwdArea	String	<p>Set password area</p> <p>K=Kill,</p> <p>W=Write (Default),</p>

		R=Read, S=Status
action	String	S=Set Password
newPwd	String	New password for password area setting above · 8 HEX characters. (0~9, A,B,C,D,E,F)
writePwd	String	Writing password · 8 HEX characters. (0~9, A,B,C,D,E,F)
killTag_Action method		
Description	Kill UHF GJB tag.	
Syntax	killTag_Action: function (killPwd)	
Parameter	Type	Description
killPwd	String	Killing password · 8 HEX characters. (0~9, A,B,C,D,E,F)
printBMP method		
Description	Convert a color image to Bitmap and directly write it into the drawing area memory.	
Syntax	printBMP(x, y, byteWidth, dotHeight, mode, filename, fileData, threshold)	

Parameter	Type	Description
x	int	The x-coordinate of the starting point (in dots)
y	int	The y-coordinate of the starting point (in dots)
byteWidth	int	Image width (in bytes)
dotHeight	int	Image height (in dots)
mode	int	Graphic modes listed below: 0: OVERWRITE 1: OR 2: XOR
filename	String	Image name
fileData	Buffer	image buffer data;
threshold	int	Convert a grayscale image to a 1-bit depth BMP image using a specified threshold. Range: 0~255, typically set to 128.
printBMP_Compression method		
Description	Convert a color image to Bitmap, compress it for transmission, and write it into the memory of the drawing area.	

Syntax	printBMP_Compression(x, y, byteWidth, dotHeight,, filename, fileData, threshold)	
Parameter	Type	Description
x	int	The x-coordinate of the starting point (in dots)
y	int	The y-coordinate of the starting point (in dots)
byteWidth	int	Image width (in bytes)
dotHeight	int	Image height (in dots)
filename	String	Image name
fileData	Buffer	image buffer data;
threshold	int	Convert a grayscale image to a 1-bit depth BMP image using a specified threshold. Range: 0~255, typically set to 128.

Appendix

Code Type	Description	Narrow : Width					Max. data length
		1:1	1:2	1:3	2:5	3:7	
128	Code 128, switching code subset automatically.	V					
128M	Code 128, switching code subset manually.	V					
EAN128	EAN128, switching code subset automatically.	V					
EAN128M	EAN128M, switching code subset manually.	V					
25	Interleaved 2 of 5.		V	V	V		Length is even
25C	Interleaved 2 of 5 with check digit.		V	V	V		Length is odd
25S	Standard 2 of 5.		V	V	V		
25I	Industrial 2 of 5.		V	V	V		
39	Code 39, switching standard and full ASCII mode automatically.		V	V	V		
39C	Code 39 with check digit.		V	V	V		
93	Code 93.			V			
EAN13	EAN 13.	V					12
EAN13+2	EAN 13 with 2 digits add-on.	V					14
EAN13+5	EAN 13 with 5 digits add-on.	V					17
EANB	EAN 8.	V					7
EANB+2	EAN 8 with 2 digits add-on.	V					96
EANB+5	EAN 8 with 5 digits add-on.	V					12
CODA	Codabar.		V	V	V		
POST	Postnet.	V					5,9,11
UPCA	UPC-A.	V					11
UPCA+2	UPC-A with 2 digits add-on.	V					13
UPA+5	UPC-A with 5 digits add-on.	V					16
UPCE	UPC-E.	V					6
UPCE+2	UPC-E with 2 digits add-on.	V					8
UPE+5	UPC-E with 5 digits add-on.	V					11
MSI	MSI.		V	V	V		
MSIC	MSI with check digit.		V	V	V		

PLESSEY	PLESSEY.		V	V	V		
CPOST	China post.					V	
ITF14	ITF14.		V	V	V		13
EAN14	EAN14.	V					13
11	Code 11.		V	V	V		
TELEPEN	Telepen. *Since V6.89EZ.		V	V	V		
TELEPENNN	Telepen number. *Since V6.89EZ.		V	V	V		
PLANET	Planet. *Since V6.89EZ.	V					
CODE49	Code 49. *Since V6.89EZ.	V					
DPI	Deutsche Post Identcode. *Since V6.91EZ.		V	V	V		11
DPL	Deutsche Post Leitcode. *Since V6.91EZ.		V	V	V		13
LOGMARS	A special use of Code 39. *Since V6.88EZ.		V	V	V		

Appednix_II

RFID Read Error Code

Error Code	Description
1	RFID module initial failure
2	Read error
3	command parameter error
5	Time Out Error
6	Module response error
100	Other error
101	Out of memory
102	Memory locked
103	receive power too weak

104	Non- specific error
105	CRC error
106	RFID transmit Error · response how many number of written words.
107	RFID receive Error · response how many number of written words.
108	No label detected
109	Command format error
110	Power level setting failure
111	Regulation setting failure