

2022

GoDEX Ezio.so Linux DLL User Manual

Control the GoDEX printer through the
program

This manual will describe how to use Ezio DLL to control GoDEX Printer.
Through simple examples, you will learn how to setup the printer parameters,
print text, images and barcodes



Directory

Directory	1
A. Overview.....	3
B. List of Functions	4
C. Description of The Functions and Parameters.....	5
1. Bar	5
2. Bar_S	6
3. Bar_QRcode	7
4. Bar_QRcode_S	8
5. Close	8
6. closeport	9
7. closeUSB	9
8. DownloadFont_TextOut.....	9
9. DownloadFont_TextOut_S.....	10
10. DrawHorLine	11
11. DrawVerLine	11
12. ecTextOut.....	12
13. ecTextOutR	13
14. ecTextOutRW	14
15. ecTextOutW	14
16. InternalFont_TextOut.....	15
17. InternalFont_TextOut_S	16
18. Open	17
19. OpenDriver.....	17
20. openUSB	17
21. RcvBuf	17
22. sendbuf	18
23. sendcommand.....	18

GoDEX Ezio.so Linux DLL User Manual

24.	setup	18
25.	TrueTypeFont_TextOut	19
26.	TrueTypeFont_TextOut_S	21
D.	How to Get The Printer Status.....	21
E.	Reference	22
i.	Reference I	22
ii.	Reference II.....	23

A. Overview

EZio. so is a Dynamic Link Library provided by GoDEX. Developers can use this component to develop control programs for printers on the Linux operating system. This device is recommended for use on Linux operating systems.

EZio.so provides control functions for USB communication. Developers must first call the `openUSB()` function to establish communication between the program and the printer before starting to give instructions to the printer; after setting or printing, they must call the `closeUSB()` function to end the connection between the program and the printer.

Developers can send instructions or data to the printer through the `sendcommand()` or `sendbuf()` function; they can get the data sent by the printer through the `RcvBuf()` function. Developers must use the EZPL printer language to control GoDEX printers, if you need more advanced control methods, you can go to [the GoDEX official website](#) to download EZPL Programming Manual.

For example, the EZPL prints a self-test page with an instruction of `~V`. If you want to print through the program, you must `sendcommand("~V")` in the program;

Developers can use the `ecTextOut()` correlation function to print text and the `Bar()` correlation function to print barcodes.

Finally, it must be noted that the EZPL printer language has a rule when [printing]: the printed content must be surrounded by `^L` and `E`, and it is not necessary if it is an instruction other than printing.

Example:

```
openUSB();
sendcommand("^L");
:
sendcommand(command);
:
sendcommand("E");
closeUSB();
```

B.List of Functions

NO.	Command	Description
1	Bar	Print 1D barcodes
2	Bar_S	Print 1D barcodes (easy mode).
3	Bar_QRcode	Print the QR Code barcode
4	Bar_QRcode_S	Print QR Code barcodes (easy mode).
5	Close	End the connection to the LAN device.
6	closeport	End the connection to the appliance
7	closeUSB	End the USB printer connection
8	DownloadFont_TextOut	Print the downloaded english numerals
9	DownloadFont_TextOut_S	Print the downloaded alphanumerics (easy mode).
10	DrawHorLine	Prints the horizontal line
11	DrawVerLine	Prints vertical lines
12	ecTextOut	Print text
13	ecTextOutR	Print text (configurable rotation).
14	ecTextOutRW	Print text (Unicode text, with rotation set).
15	ecTextOutW	Print text (Unicode text).
16	InternalFont_TextOut	Print built-in or Asian characters
17	InternalFont_TextOut_S	Print built-in or Asian characters (Easy Mode).
18	Open	Start the connection to the LAN device
19	OpenDriver	Turn on the connection to the Godex CUPS device
20	openUSB	Use the USB ID to connect the printer
21	RcvBuf	Reads the data sent by the printer
22	sendbuf	Send Byte data to the printer
23	sendcommand	Send string data to the printer
24	setup	Set the basic printer parameters
25	TrueTypeFont_TextOut	Prints the downloaded vector word
26	TrueTypeFont_TextOut_S	Print the downloaded vector word (easy mode).

C. Description of The Functions and Parameters

1. Bar

Bar (CodeType, PosX, PosY, Narrow, Wide, Height, Rotation, Readable, Data)

Function description	Print 1D barcodes
Parameter definitions	<i>CodeType</i> Type : (char *). Barcode type (see Reference I). <i>PosX</i> Type : Integer The X coordinate position of the barcode (upper left horizontal position), in dots <i>PosY</i> Type : Integer The Y coordinate position of the barcode (the upper vertical position to the left), in dots <i>Narrow</i> Type : Integer The bar code is thin line width, the unit is dot, and the general setting range is 1 ~ 10 ITF14 (DUN14) has a setting range of 5 to 8 THE UPC/EAN series barcodes can be set from 2 to 4 <i>Wide</i> Type : Integer The bar code is thick and wide, in dots, and the setting range is 2 to 30 This parameter is only valid for Code 39,93, CODABAR & I 2 of 5 <i>Height</i> Type : Integer Barcode height, in dots, set range from 24 to 1200 <i>Rotation</i> Type : Integer Barcode rotation mode

GoDEX Ezio.so Linux DLL User Manual

	0) 0° , 1) 90° , 2) 180° , 3) 270° <i>Readable</i> Type : Integer The way the code text is displayed 0: Do not show 1: The bar code is placed on the left below 2: The bar code is placed on the left above 3: The bar code is centered below 4: The bar code is centered above 5: The bar code is placed on the right below 6: The bar code is placed on the right above <i>Data</i> Type : (char *). Barcode data content
The callback value	1= (integer) OK -1= (integer) FAIL

2. Bar_S

Bar_S (CodeType, PosX, PosY, Data)

Function description	Print 1D barcodes (easy mode).
Parameter definitions	<i>CodeType</i> Type : (char *). Barcode type <i>PosX</i> Type : Integer The X coordinate position of the barcode (upper left horizontal position), in dots <i>PosY</i> Type : Integer The Y coordinate position of the barcode (the upper vertical position to the left), in dots <i>Data</i> Type : (char *). Barcode data content
The callback value	1= (integer) OK -1= (integer) FAIL

3. Bar_QRcode

Bar_QRcode (PosX, PosY, Mode, Type, ErrLevel, Mask, Mul, Len, Deg, Data)

Function description	Print the QR Code barcode
Parameter definitions	<i>PosX</i> Type : Integer The X coordinate position of the barcode (upper left horizontal position), in dots
	<i>PosY</i> Type : Integer The Y coordinate position of the barcode (the upper vertical position to the left), in dots
	<i>Mode</i> Type : Integer Encoding mode 1 : Numerical data mode 2 : Alpha numerical data mode 3 : 8-bit data mode 4 : Kanji data mode 5: Mixing mode (not available for Micro QR Code).
	<i>Type</i> Type : Integer Set the type 1 : Model1 (original) 2 : Model2 (enhanced) 3 : Micro QR Code
	<i>ErrLevel</i> Type : (char *). Error correction level L : Low M : Medium Q : Medium high H: High (not available for Micro QR Code).
	<i>Mask</i> Type : Integer Shielding factor, set from 0 to 7. Or set to 8 to switch automatically For Micro QR Code, this value must be set to 0
	<i>Mul</i>

GoDEX Ezio.so Linux DLL User Manual

	Type : Integer Magnification, set from 1 to 40 <i>Len</i> Type : Integer The length of the data <i>Deg</i> Type : Integer Barcode rotation mode 0) 0° , 1) 90° , 2) 180° , 3) 270° <i>Data</i> Type : (char *). Barcode data content
The callback value	1= (integer) OK -1= (integer) FAIL

4. Bar_QRcode_S

Bar_QRcode_S (PosX, PosY, Len, Data)

Function description	Print QR Code barcodes (easy mode).
Parameter definitions	<i>PosX</i> Type : Integer The X coordinate position of the barcode (upper left horizontal position), in dots <i>PosY</i> Type : Integer The Y coordinate position of the barcode (the upper vertical position to the left), in dots <i>Len</i> Type : Integer The length of the data <i>Data</i> Type : (char *). Barcode data content
The callback value	1= (integer) OK -1= (integer) FAIL

5. Close

Close()

GoDEX Ezio.so Linux DLL User Manual

Function description	End the connection to the LAN device.
Parameter definitions	None
The callback value	1 = (integer) OK 0 = (integer) FAIL

6. **closeport**

closeport ()

Function description	End the connection to the appliance
Parameter definitions	None
The callback value	0 = (integer) OK -1 = (integer) FAIL

7. **closeUSB**

closeUSB ()

Function description	End the USB printer connection
Parameter definitions	None
The callback value	None

8. **DownloadFont_TextOut**

DownloadFont_TextOut (FontName, PosX, PosY, Mul_X, Mul_Y, Gap, RotationInverse, Data)

Function description	Print the downloaded alphanumerics (the alphanumerics must already be downloaded within the printer).
Parameter definitions	<i>FontName</i> Type : (char *) Font pronouns. The range can be set from A to Z (The name must correspond to the downloaded font). <i>PosX</i> Type : Integer The X-coordinate position of the text (upper-left

	horizontal position), in dots <i>PosY</i> Type : Integer The Y-coordinate position of the text (the upper-left vertical position), in dot <i>Mul_X</i> Type : Integer Horizontal magnification (up to 8x). <i>Mul_Y</i> Type : Integer Vertical magnification (up to 8x). <i>Gap</i> Type : Integer The white space spacing between characters can be set from 0 to 200 dots <i>RotationInverse</i> Type : (char *) The built-in word setting range is 0~3, and the Asian word setting range is 0~7 0) 0° , 1) 90° , 2) 180° , 3) 270° 4) 0° , 5) 90° , 6) 180° , 7) 270° 0~3 rotates all characters in the string together 4~7 is rotated separately for characters When the character "I" is added to the end of this parameter, it can be printed in reverse black and white <i>Data</i> Type : (char *) String data
The callback value	1= (integer) OK -1= (integer) FAIL

9. DownloadFont_TextOut_S

DownloadFont_TextOut_S (FontName, PosX, PosY, Data)

Function description	Print the downloaded alphanumerics (easy mode).
Parameter definitions	<i>FontName</i> Type : (char *) Font pronouns, which can be set to A ~ Z (The name must correspond to the downloaded font).

	<i>PosX</i> Type : Integer The X-coordinate position of the text (upper-left horizontal position), in dots <i>PosY</i> Type : Integer The Y-coordinate position of the text (the upper-left vertical position), in dot <i>Data</i> Type : (char *) String data
The callback value	1= (integer) OK -1= (integer) FAIL

10. DrawHorLine

DrawHorLine (PosX, PosY, Length, Thick)

Function description	Prints the horizontal line
Parameter definitions	<i>PosX</i> Type : Integer The X coordinate of the start point in the upper-left corner of the horizontal line, in dot <i>PosY</i> Type : Integer The Y coordinate of the starting point of the upper-left corner of the horizontal line, in dot <i>Length</i> Type : Integer Horizontal line length, in dots <i>Thick</i> Type : Integer The line width (thickness setting) of the horizontal line, in dots
The callback value	1= (integer) OK -1= (integer) FAIL

11. DrawVerLine

DrawVerLine (PosX, PosY, Length, Thick)

GoDEX Ezio.so Linux DLL User Manual

Function description	Prints vertical lines
Parameter definitions	<i>PosX</i> Type : Integer The X coordinate of the start point in the upper-left corner of the vertical line, in dot <i>PosY</i> Type : Integer The Y coordinate at the beginning of the upper left corner of the vertical line, in dots <i>Length</i> Type : Integer Vertical line length, in dots <i>Thick</i> Type : Integer The line width (thickness setting) of the vertical line, in dots
The callback value	1= (integer) OK -1= (integer) FAIL

12. **ecTextOut**

ecTextOut(PosX, PosY, Height, FontName, Data)

Function description	Print text
Parameter definitions	<i>PosX</i> Type : Integer The X-coordinate position of the text (upper-left horizontal position), in dots <i>PosY</i> Type : Integer The Y-coordinate position of the text (the upper-left vertical position), in dot <i>Height</i> Type : Integer Word height <i>FontName</i> Type : (char *). The name of the font <i>Data</i>

GoDEX Ezio.so Linux DLL User Manual

	Type : (char *). String data content
The callback value	1= (integer) OK -1= (integer) FAIL

13. **ecTextOutR**

ecTextOutR(PosX, PosY, Height, FontName, Data, Width, Degree)

Function description	Print text (configurable rotation).
Parameter definitions	<i>PosX</i> Type : Integer The X-coordinate position of the text (upper-left horizontal position), in dots <i>PosY</i> Type : Integer The Y-coordinate position of the text (the upper-left vertical position), in dot <i>Height</i> Type : Integer Word height <i>FontName</i> Type : (char *). The name of the font <i>Data</i> Type : (char *). String data content <i>Width</i> Type : Integer Word width <i>Degree</i> Type : Integer Rotation angle 0: degree 0 90: degree90 180 : degree 180 270 : degree 270
The callback value	1= (integer) OK -1= (integer) FAIL

14. ecTextOutRW

ecTextOutRW(PosX, PosY, Height, FontName, Data, Width, Deg)

Function description	Print text (Unicode text, with rotation set).
Parameter definitions	<i>PosX</i> Type : Integer The X-coordinate position of the text (upper-left horizontal position), in dots <i>PosY</i> Type : Integer The Y-coordinate position of the text (the upper-left vertical position), in dot <i>Height</i> Type : Integer Word height <i>FontName</i> Type : (char *). The name of the font <i>Data</i> Type : (char *). String data content <i>Width</i> Type : Integer Word width <i>Degree</i> Type : Integer Rotation angle 0: degree 0 90: degree90 180 : degree 180 270 : degree 270
The callback value	1= (integer) OK -1= (integer) FAIL

15. ecTextOutW

ecTextOutW(PosX, PosY, Height, FontName, Data)

Function description	Print text (Unicode text).
Parameter	<i>PosX</i>

definitions	Type : Integer The X-coordinate position of the text (upper-left horizontal position), in dots <i>PosY</i> Type : Integer The Y-coordinate position of the text (the upper-left vertical position), in dot <i>Height</i> Type : Integer Word height <i>FontName</i> Type : (char *). The name of the font <i>Data</i> Type : (char *). String data content
The callback value	1= (integer) OK -1= (integer) FAIL

16. InternalFont_TextOut

InternalFont_TextOut (FontType, PosX, PosY, Mul_X, Mul_Y, Gap, RotationInverse, Data)

Function description	Print built-in or Asian characters (Asian characters must already be downloaded from the printer).
Parameter definitions	<i>FontType</i> Type : (char *) Font type (see Reference II). <i>PosX</i> Type : Integer The X-coordinate position of the text (upper-left horizontal position), in dots <i>PosY</i> Type : Integer The Y-coordinate position of the text (the upper-left vertical position), in dot <i>Mul_X</i> Type : Integer Horizontal magnification (up to 8x). <i>Mul_Y</i>

GoDEX Ezio.so Linux DLL User Manual

	Type : Integer Vertical magnification (up to 8x). <i>Gap</i> Type : Integer The white space spacing between characters can be set from 0 to 200 dots <i>RotationInverse</i> Type : (char *) The built-in word setting range is 0~3, and the Asian word setting range is 0~7 0) 0° , 1) 90° , 2) 180° , 3) 270° 4) 0° , 5) 90° , 6) 180° , 7) 270° 0~3 rotates all characters in the string together 4~7 is rotated separately for characters When the character "I" is added to the end of this parameter, it can be printed in reverse black and white <i>Data</i> Type : (char *) String data
The callback value	1= (integer) OK -1= (integer) FAIL

17. InternalFont_TextOut_S

InternalFont_TextOut_S (FontType, PosX, PosY, Data)

Function description	Print built-in or Asian characters (Easy Mode).
Parameter definitions	<i>FontType</i> Type : (char *) Font type (see Reference II). <i>PosX</i> Type : Integer The X-coordinate position of the text (upper-left horizontal position), in dots <i>PosY</i> Type : Integer The Y-coordinate position of the text (the upper-left vertical position), in dot <i>Data</i> Type : (char *)

	String data
The callback value	1= (integer) OK -1= (integer) FAIL

18. Open

Open(IP, Port)

Function description	Start the connection to the LAN device
Parameter definitions	<i>IP</i> Type : const char* ex : 192.168.102.1 <i>Port</i> Type : Int
The callback value	1 = (integer) OK 0 = (integer) FAIL

19. OpenDriver

OpenDriver(Port)

Function description	Turn on the connection to the Godex CUPS device
Parameter definitions	<i>Port</i> Type : (char*) The path to the CUPS device
The callback value	0 = (integer) OK -1= (integer) FAIL

20. openUSB

openUSB()

Function description	Use the USB ID to connect the printer
Parameter definitions	None
The callback value	1= (integer) OK -1= (integer) FAIL

21. RcvBuf

RcvBuf(Buf, Len)

GoDEX Ezio.so Linux DLL User Manual

Function description	Reads the data sent by the printer
Parameter definitions	<i>Buf</i> Type : (unsigned char *). The data that the printer calls back <i>Len</i> Type : Integer The size of the array of characters receiving data in the program (that is, the size of Buf's array).
The callback value	The length of the printer's postback data -1= (integer) FAIL

22. **sendbuf**

sendbuf(Data, Len)

Function description	Send Byte data to the printer
Parameter definitions	<i>Data</i> Type : (char *). Byte data transferred to the printer <i>Len</i> Type : Integer The length of the data to be transmitted
The callback value	The length of the printer's postback data -1= (integer) FAIL

23. **sendcommand**

sendcommand(command)

Function description	Send string data to the printer
Parameter definitions	<i>command</i> Type : (char *). String data transmitted to the printer
The callback value	1= (integer) OK -1= (integer) FAIL

24. **setup**

setup(height, dark, speed, mode, gap, top)

GoDEX Ezio.so Linux DLL User Manual

Function description	Set the basic printer parameters
Parameter definitions	<i>height</i> Type : Integer Paper height in mm <i>dark</i> Type : Integer Printing blackness can be set from 0 to 19 <i>speed</i> Type : Integer Print speed, in IPS, can be set from 2 to 7 There are different range limits depending on the printer <i>mode</i> Type : Integer Paper type 0: Label with gap 1: Plain paper 2 : Black mark label <i>gap</i> Type : Integer Label spacing, in mm <i>top</i> Type : Integer The black line marks the distance in mm
The callback value	1= (integer) OK -1= (integer) FAIL

25. TrueTypeFont_TextOut

TrueTypeFont_TextOut (FontName, PosX, PosY, Font_W, Font_H, SpaceChar, RotationInverse, TTFTable, WidthMode, Data)

Function description	Prints the downloaded vector word (the vector word must already be downloaded within the printer).
Parameter definitions	<i>FontName</i> Type : (char *) Font pronouns. The range can be set from A to Z (The name must correspond to the downloaded font). <i>PosX</i> Type : Integer The X-coordinate position of the text (upper-left

	horizontal position), in dots
<i>PosY</i>	Type : Integer The Y-coordinate position of the text (the upper-left vertical position), in dot
<i>Font_W</i>	Type : Integer The magnification of the font width scaling can be set from 8 to 2000 dots
<i>Font_H</i>	Type : Integer The magnification of the font height scaling can be set from 8 to 2000 dots
<i>SpaceChar</i>	Type : Integer The white space spacing between characters can be set from 0 to 200 dots
<i>RotationInverse</i>	Type : (char *) The built-in word setting range is 0~3, and the Asian word setting range is 0~7 0) 0° , 1) 90° , 2) 180° , 3) 270° 4) 0° , 5) 90° , 6) 180° , 7) 270° 0~3 rotates all characters in the string together 4~7 is rotated separately for characters When the character "I" is added to the end of this parameter, it can be printed in reverse black and white
<i>TTFTable</i>	Type : (char *) The corresponding table for the setting vector word lookup (if the setting is wrong, the wrong text will be printed). Set to 0: Use the built-in ASCII counterpart table If set to A~Z: Use the user-downloaded Unicode counterpart table
<i>WidthMode</i>	Type : Integer 0 : Width/height ratio mode 1 : Monospace mode (not open yet).
<i>Data</i>	

	Type : (char *) String data
The callback value	1= (integer) OK -1= (integer) FAIL

26. TrueTypeFont_TextOut_S

TrueTypeFont_TextOut_S (FontName, PosX, PosY, Data)

Function description	Print the downloaded vector word (easy mode).
Parameter definitions	<i>FontName</i> Type : (char *) Font pronouns, which can be set to A ~ Z (The name must correspond to the downloaded font). <i>PosX</i> Type : Integer The X-coordinate position of the text (upper-left horizontal position), in dots <i>PosY</i> Type : Integer The Y-coordinate position of the text (the upper-left vertical position), in dot <i>Data</i> Type : (char *) String data
The callback value	1= (integer) OK -1= (integer) FAIL

D. How to Get The Printer Status

If you want to know the status of the printer, you can use sendcommand to send the command ~S first, CHECK and then use RcvBuf to read the status number of the postback.

~S,CHECK - Status immediate response command

Syntax	~S,CHECK								
Parameter	None								
Description	The command will show the status of printer in "aa <CR><LF>" format. aa = printer status information: 00 – Ready 01 – Media Empty or Media Jam 02 – Media Empty or Media Jam 03 – Ribbon Empty 04 – Printhead Up (Open) 05 – Rewinder Full 06 – File System Full 07 – Filename Not Found 08 – Duplicate Name 09 – Syntax error 10 – Cutter JAM 11 – Extended Memory Not Found 20 – Pause 21 – In Setting Mode 22 – In Keyboard Mode 50 – Printer is Printing 60 – Data in Process *Note: Before using this command, the "^XSET,IMMEDIATE" (Set immediate response on/off) command should be turned on.								
Example	Procedure: <table border="1"> <tr> <td>PC to Printer</td><td>^XSET,IMMEDIATE,1</td></tr> <tr> <td>Printer action</td><td>Door open</td></tr> <tr> <td>PC to Printer</td><td>~S,CHECK</td></tr> <tr> <td>Printer to PC</td><td>04</td></tr> </table> **04: Door open	PC to Printer	^XSET,IMMEDIATE,1	Printer action	Door open	PC to Printer	~S,CHECK	Printer to PC	04
PC to Printer	^XSET,IMMEDIATE,1								
Printer action	Door open								
PC to Printer	~S,CHECK								
Printer to PC	04								

E.Reference

i. Reference I

Code Type definition for one-dimensional barcodes

Code Type	Description	Code Type	Description
A	Code 39 STD	O	Codabar
A2	Code 39 STD w check	P	Code 93
A3	Code 39 full ASCII	Q	Code 128 (auto)
A4	Code 39 full ASCII w	Q2	Code 128 (subset A/B/C)
A5	Code 39 STD w check	R	UCC 128
A6	Code 39 STD w *	S	Post NET
B	EAN 8	T	UPC Interleaved 2 of 5
C	EAN 8 - Add ON 2	U	EAN 128
D	EAN 8 - Add ON 5	V	RPS 128
E	EAN 13	X	HIBC
F	EAN 13 – Add ON 2	Y	MSI 1 MOD 10

GoDEX Ezio.so Linux DLL User Manual

G	EAN 13 – Add ON 5	Y2	MSI 2 MOD 10
H	UPC A	Y3	MSI 1 MOD 11 & 10
I	UPC A - Add ON 2	Y4	MSI no digit check
J	UPC A - Add ON 5	Z	I 2 of 5 with Shipping Bearer Bars
K	UPC E	1	UCC/EAN-128 K-MART
L	UPC E - Add ON 2	2	UCC/EAN-128 RANDOM
M	UPC E - Add ON 5	3	Telepen
N	I 2 of 5	4	FIM
N2	I 2 of 5 with check digit	7	Plessey

ii. Reference II

Built-in font size definitions (TTF Fonts, Code page 850).

Font Type	200 DPI	300 DPI	600 DPI
A	6 Points	4 Points	2 Points
B	8 Points	5 Points	2.5 Points
C	10 Points	6 Points	3 Points
D	12 Points	8 Points	4 Points
E	14 Points	9 Points	4.5 Points
F	18 Points	12 Points	6 Points
G	24 Points	16 Points	8 Points
H	30 Points	20 Points	10 Points
I	16 x 26 Dots	24 Points	12 Points
J	-	10 x 17 Dots	10 x 17 Dots
Zn, n = 1 ~ 4	Asia font		

END