

2020

Fabulous

动态库开发手册

For Fabulous Printer

此手册中将介绍如何透过 FBio DLL 来撰写 Fabulous 打印机控制程序。透过简单的范例，您将学习到如何设定打印机参数，打印文字，影像与条形码

V1.0.0.11

Fabulous

2020/8/8



目录

1	概观说明.....	3
2	函式清单.....	4
3	函式参数说明.....	7
3.1	openport	7
3.2	OpenDriver.....	7
3.3	FindFirstUSB	7
3.4	FindNextUSB	8
3.5	OpenUSB	8
3.6	FindFirstNet.....	8
3.7	FindNextNet	8
3.8	OpenNet.....	9
3.9	setbaudrate	9
3.10	closeport	9
3.11	setup	10
3.12	sendcommand	10
3.13	sendbuf	11
3.14	RcvBuf	11
3.15	intloadimage	11
3.16	extloadimage.....	12
3.17	downloadimage	12
3.18	putimage	13
3.19	putimage_Halftone	14
3.20	ecTextOut	14
3.21	ecTextOutW.....	15
3.22	ecTextOutR.....	16
3.23	ecTextOutRW	17
3.24	ecTextOutFine	18
3.25	ecTextOutFineW.....	19
3.26	ecTextDownLoad	21
3.27	ecTextDownLoadW	22
3.28	Bar	23
3.29	Bar_S	24
3.30	Bar_GS1DataBar.....	25
3.31	Bar_GS1DataBar_S.....	26
3.32	Bar_PDF417.....	27
3.33	Bar_PDF417_S.....	28

Fabulous FBio DLL User Manual

3.34	Bar_MicroPDF417	29
3.35	Bar_MicroPDF417_S	30
3.36	Bar_Maxicode	31
3.37	Bar_Maxicode_S	32
3.38	Bar_DataMatrix.....	33
3.39	Bar_DataMatrix_S.....	34
3.40	Bar_QRcode	35
3.41	Bar_QRcode_S	36
3.42	Bar_Aztec	37
3.43	Bar_Aztec_S	38
3.44	InternalFont_TextOut.....	39
3.45	InternalFont_TextOut_S.....	40
3.46	DownloadFont_TextOut	41
3.47	DownloadFont_TextOut_S	42
3.48	TrueTypeFont_TextOut.....	43
3.49	TrueTypeFont_TextOut_S.....	44
3.50	DrawHorLine	45
3.51	DrawVerLine.....	45
3.52	FillRec	46
3.53	DrawRec	46
3.54	DrawOblique	47
3.55	DrawEllipse.....	47
3.56	DrawRoundRec.....	48
3.57	DrawTriangle	49
3.58	DrawDiamond	49
3.59	GetDllVersion	50
4.	程序开发指引.....	51
4.1	C# 程序开发.....	52
4.2	VB.NET 程序开发	52
4.3	BCB6 程序开发.....	52
4.4	Delphi 程序开发.....	53
4.5	VC (MFC) 程序开发.....	53
4.6	Java 程序开发	53
	附件一 (一维条形码型态).....	54
	附件二 (内建字型态).....	55

1 概观说明

FBio DLL 是由 Fabulous 提供的动态链接数据库(Dynamic Link Library)。程序开发者可使用此组件在微软 Windows 操作系统上开发打印机的控制程序。目前 FBioDLL 提供 32 位与 64 位 2 个版本，用户可以根据开发环境的设定，选用不同的 DLL。此组件建议在 Windows XP 以上的操作系统使用，目前的最新版本可支持至 Windows 8。

FBio DLL 提供了 USB、COM、LPT、Driver 与网络等通讯的控制函式。开发者在开始对打印机下指令前，必须先呼叫 `openport()`、`OpenUsb()`、`OpenNet()` 或 `OpenDriver()` 其中之一个函式来建立程序与打印机间的通讯；在设定或打印完毕后，必须呼叫 `closeport()` 函式来结束程序与打印机间的联机。

开发者可以透过 `sendcommand()` 或 `sendbuf()` 函式，传送指令或数据给打印机；可透过 `RcvBuf()` 函式取得打印机送出的数据。开发者必须使用 EZPL 打印机语言来控制 Fabulous 打印机。

举例来说：EZPL 印出自我测试页的指令为 `~V`。如果要透过程序行印，就必须在程序中送出 `sendcommand("~V");`

开发者可以使用 `ecTextOut()` 相关函式来打印文字，使用 `putimage()` 函式来打印图片，使用 `Bar()` 相关函式来打印条形码。

最后必须注意的是，EZPL 打印机语言在【打印】时有一个规则：打印的内容必须以 `^L` 与 `E` 包围住，如果是打印以外的指令则不必。

```
openport(6);  
sendcommand("%L")  
; :  
sendcommand(指令内容);  
:  
sendcommand("%E");  
closeport();
```

2 函式清单

NO.	Command	Description
1	openport	开启 USB、COM、LPT 通讯接口，与打印机建立联机 (此函式仅支持特定通讯接口)
2	OpenDriver	使用打印机的 Driver 名称建立联机
3	FindFirstUSB	取得第一个已联机的 USB 打印机 ID 若回传 false 则表示没有已联机的 USB 打印机
4	FindNextUSB	取得下一个已联机的 USB 打印机 ID 若回传 false 则表示没有其他的 USB 打印机
5	OpenUsb	指定 USB ID 开启打印机联机
6	FindFirstNet	搜寻区网中第一个网络打印机 IP 与端口 若回传 false 则表示没有搜寻到网络打印机
7	FindNextNet	搜寻区网中下一个网络打印机 IP 与端口 若回传 false 则表示没有搜寻到其他打印机
8	OpenNet	指定 IP 与端口开启打印机联机 固定 IP 的打印机，可使用此函式建立联机
9	setbaudrate	如果通讯界面选择 COM，可以设定此接口的 Baud Rate (如果不设定，预设为 9600)
10	closeport	结束打印机联机。
11	setup	设定标签纸种类、纸张高度、打印黑度、打印速度等参数。
12	sendcommand	传送字符串数据给打印机
13	sendbuf	送出 Byte 数据给打印机
14	RcvBuf	读取打印机送出的数据
15	intloadimage	下载 PCX 或 BMP 格式的单色调图像文件至打印机的内部存储器(EEPROM)。 ps. 必须下^Y 指令才会打印
16	extloadimage	下载 PCX 或 BMP 格式的单色调图像文件至打印机的外部内存(SDRAM)。 ps. 必须下^Y 指令才会打印
17	downloadimage	下载 BMP 或 JPG 格式的多色调影像至打印机的外部内存(SDRAM)，并对图形做半色调处理。 ps. 必须下^Y 指令才会打印

Fabulous FBio DLL User Manual

18	<u>putimage</u>	直接打印 BMP 或 JPG 格式的多色调影像 (不需下载图文件至打印机)
19	<u>putimage_Halftone</u>	直接打印 BMP 或 JPG 格式的多色调影像, 并可选择半色调处理方式
20	<u>ecTextOut</u>	打印文字
21	<u>ecTextOutW</u>	打印文字 (Unicode 文字)
22	<u>ecTextOutR</u>	打印文字 (可设定旋转)
23	<u>ecTextOutRW</u>	打印文字 (Unicode 文字, 可设定旋转)
24	<u>ecTextOutFine</u>	打印文字 (可设定旋转, 黑度, 斜体, 底线, 删除线, 颜色反向)
25	<u>ecTextOutFineW</u>	打印文字 (Unicode 文字, 可设定旋转, 黑度, 斜体, 底线, 删除线, 颜色反向)
26	<u>ecTextDownLoad</u>	将文字内容转成一个图文件, 下载至打印机中
27	<u>ecTextDownLoadW</u>	将文字(Unicode 文字)内容转成一个图文件, 下载至打印机中
28	<u>Bar</u>	打印 1D 条形码
29	<u>Bar_S</u>	打印 1D 条形码 (简易模式)
30	<u>Bar_GS1DataBar</u>	打印 GS1 DataBar 条形码
31	<u>Bar_GS1DataBar_S</u>	打印 GS1 DataBar 条形码 (简易模式)
32	<u>Bar_PDF417</u>	打印 PDF 417 条形码
33	<u>Bar_PDF417_S</u>	打印 PDF 417 条形码 (简易模式)
34	<u>Bar_MicroPDF417</u>	打印 Micro PDF 417 条形码
35	<u>Bar_MicroPDF417_S</u>	打印 Micro PDF 417 条形码 (简易模式)
36	<u>Bar_Maxicode</u>	打印 Maxicode 条形码
37	<u>Bar_Maxicode_S</u>	打印 Maxicode 条形码 (简易模式)
38	<u>Bar_DataMatrix</u>	打印 Data Matrix Code 条形码
39	<u>Bar_DataMatrix_S</u>	打印 Data Matrix Code 条形码 (简易模式)
40	<u>Bar_QRcode</u>	打印 QR Code 条形码
41	<u>Bar_QRcode_S</u>	打印 QR Code 条形码 (简易模式)
42	<u>Bar_Aztec</u>	打印 Aztec 条形码
43	<u>Bar_Aztec_S</u>	打印 Aztec 条形码 (简易模式)
44	<u>InternalFont_TextOut</u>	打印内建字或亚洲字 (打印机内必须已经先下载亚洲字)
45	<u>InternalFont_TextOut_S</u>	打印内建字或亚洲字 (简易模式)
46	<u>DownloadFont_TextOut</u>	打印下载的英数字 (打印机内必须已经先下载英数字)

Fabulous FBio DLL User Manual

47	<u>DownloadFont_TextOut_S</u>	打印下载的英数字 (简易模式)
48	<u>TrueTypeFont_TextOut</u>	打印下载的向量字 (打印机内必须已经先下载向量字)
49	<u>TrueTypeFont_TextOut_S</u>	打印下载的向量字 (简易模式)
50	<u>DrawHorLine</u>	打印水平线
51	<u>DrawVerLine</u>	打印垂直线
52	<u>FillRec</u>	打印实心矩形
53	<u>DrawRec</u>	打印空心矩形
54	<u>DrawOblique</u>	打印斜线
55	<u>DrawEllipse</u>	打印椭圆
56	<u>DrawRoundRec</u>	打印圆角矩形
57	<u>DrawTriangle</u>	打印三角形
58	<u>DrawDiamond</u>	打印菱形
59	<u>GetDllVersion</u>	取得 FBio DLL 的版本资料

3 函式参数说明

3.1 openport(PortID)

函式描述	透过指定通讯方式，建立打印机联机
参数定义	PortID 类型 : (char *) 打印机的通讯方式 “0” = LPT1 “1” = COM1 “2” = COM2 “3” = COM3 “4” = COM4 “5” = LPT2 “6” = USB
回传值	1 = (integer) OK 0 = (integer) FAIL

3.2 OpenDriver(DriverName)

函式描述	透过打印机驱动名称，建立打印机联机
参数定义	DriverName 类型 : (char *) 操作系统上 [装置和打印机] 中 Fabulous 打印机的名称
回传值	1 = (integer) OK 0 = (integer) FAIL

3.3 FindFirstUSB(UsbID)

函式描述	取得第一个已联机的 USB 打印机 ID
参数定义	UsbID 类型 : (char *) USB ID
回传值	1 = (integer) OK 0 = (integer) FAIL

3.4 FindNextUSB(UsbID)

函式描述	取得下一个已联机的 USB 打印机 ID
参数定义	UsbID 类型 : (char *) USB ID
回传值	1 = (integer) OK 0 = (integer) FAIL

3.5 OpenUSB(UsbID)

函式描述	透过 USB ID, 建立打印机联机
参数定义	UsbID 类型 : (char *) USB ID
回传值	1 = (integer) OK 0 = (integer) FAIL

3.6 FindFirstNet(IP, Port)

函式描述	搜寻区网中第一个网络打印机 IP 与端口
参数定义	IP 类型 : (char *) 打印机的 IP Address Port 类型 : (char *) 打印机的网络端口
回传值	1 = (integer) OK 0 = (integer) FAIL

3.7 FindNextNet(IP, Port)

函式描述	搜寻区网中下一个网络打印机 IP 与端口
参数定义	IP 类型 : (char *) 打印机的 IP Address Port 类型 : (char *) 打印机的网络端口
回传值	1 = (integer) OK 0 = (integer) FAIL

3.8 OpenNet(IP, Port)

函式描述	指定 IP 与端口开启打印机联机
参数定义	<p>IP</p> <p>类型 : (char *)</p> <p>打印机的 IP Address</p> <p>Port</p> <p>类型 : (char *)</p> <p>打印机的网络端口</p>
回传值	<p>1 = (integer) OK</p> <p>0 = (integer) FAIL</p>

3.9 setbaudrate(Baudrate)

函式描述	设定 COM(RS232)通讯的 Baud Rate
参数定义	<p>Baudrate</p> <p>类型 : Integer</p> <p>COM(RS232)的 Baud Rate</p> <p>Baud Rate 必须配合打印机的设定，若有错误会无法通讯</p>
回传值	None

3.10 closeport()

函式描述	结束打印机联机
参数定义	None
回传值	None

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

函式描述	设定打印机基本参数
参数定义	<p>height 类型 : Integer 纸张高度, 单位为 mm</p> <p>dark 类型 : Integer 打印黑度, 可设定范围为 0~19</p> <p>speed 类型 : Integer 打印速度, 单位为 IPS, 可设定范围为 2~7 根据不同打印机有不同的范围限制</p> <p>mode 类型 : Integer 纸张类型 0: 标签纸 (label with gap) 1: 连续纸 (plain paper) 2: 黑线纸 (black mark label)</p> <p>gap 类型 : Integer 卷标间距, 单位为 mm</p> <p>top 类型 : Integer 黑线标记距离, 单位为 mm</p>
回传值	<p>1 = (integer) OK</p> <p>0 = (integer) FAIL</p>

3.12 **sendcommand**(command)

函式描述	传送字符串数据给打印机
参数定义	<p>command 类型 : (char *) 传送给打印机的字符串数据</p>
回传值	<p>1 = (integer) OK</p> <p>0 = (integer) FAIL</p>

3.13 sendbuf(Data, Len)

函式描述	传送 Byte 数据给打印机
参数定义	Data 类型 : (char *) 传送给打印机的 Byte 数据 Len 类型 : Integer 欲传送数据的长度
回传值	回传值为已传送成功的数据长度。如果值为 0 表示没有任何数据被送至打印机

3.14 RcvBuf(Buf, Len)

函式描述	读取打印机送出的数据
参数定义	Buf 类型 : (unsigned char *) 打印机回传的数据 Len 类型 : Integer 程序中收数据的字符数组的大小 (即 Buf 的数组大小)
回传值	打印机回传数据的长度

3.15 intloadimage(Filename, ID_Name, Image_Type)

函式描述	下载单色调图像文件至打印机的内部存储器
参数定义	Filename 类型 : (char *) 要上传至打印机影像的文件名位置 ID_Name 类型 : (char *) 上传后打印图文件指令(^Y)要呼叫的名称 Image_Type 类型 : (char *) 图档格式 pcx : for .pcx image file bmp : for .bmp image file
回传值	1 = (integer) OK 0 = (integer) FAIL

3.16 extloadimage(Filename, ID_Name, Image_Type)

函式描述	下载单色调图像文件至打印机的外部内存
参数定义	<p>Filename 类型 : (char *) 要上传至打印机影像的文件名位置</p> <p>ID_Name 类型 : (char *) 上传后打印图文件指令(^Y)要呼叫的名称</p> <p>Image_Type 类型 : (char *) 图档格式 pcx : pcx 类型的单色调档案 bmp : bmp 类型的单色调档案</p>
回传值	<p>1 = (integer) OK</p> <p>0 = (integer) FAIL</p>

3.17 downloadimage(Filename, Degree, ID_Name)

函式描述	下载多色调影像至打印机的外部内存，并做半色调处理
参数定义	<p>Filename 类型 : (char *) 要上传至打印机影像的文件名位置</p> <p>Degree 类型 : Integer 0 : 无旋转 90 : 旋转 90 度</p> <p>ID_Name 类型 : (char *) 上传后打印图文件指令(^Y)要呼叫的名称</p>
回传值	<p>1 = (integer) OK</p> <p>0 = (integer) FAIL</p>

3.18 putimage(PosX, PosY, Filename, Degree)

函式描述	直接打印 BMP 或 JPG 格式的多色调影像, 套用聚焦式处理 (不需下载图文件至打印机)
参数定义	<p>PosX 类型 : Integer 图形的 X 坐标位置(左上方水平位置), 单位为 dot</p> <p>PosY 类型 : Integer 图形的 Y 坐标位置(左上方垂直位置), 单位为 dot</p> <p>Filename 类型 : (char *) 要上传至打印机影像的文件名位置</p> <p>Degree 类型 : Integer 0: 无旋转, 90: 旋转 90 度</p>
回传值	<p>1 = (integer) OK</p> <p>0 = (integer) FAIL</p>

3.19 putimage_Halftone(PosX, PosY, Filename, Degree, Halftone)

函式描述	直接打印 BMP 或 JPG 格式的多色调影像, 并可选择半色调处理方式 (不需下载图文件至打印机)
参数定义	<p>PosX 类型 : Integer 图形的 X 坐标位置(左上方水平位置), 单位为 dot</p> <p>PosY 类型 : Integer 图形的 Y 坐标位置(左上方垂直位置), 单位为 dot</p> <p>Filename 类型 : (char *) 要上传至打印机影像的文件名位置</p> <p>Degree 类型 : Integer 0: 无旋转, 90: 旋转 90 度</p> <p>Halftone Type : Integer 0: 无特殊处理 (None) 1: 聚焦式处理 (Cluster Dithering) 2: 分布式处理 (Dispersed Dithering) 3: 扩散式处理 (Diffusion Dithering)</p>
回传值	<p>1 = (integer) OK</p> <p>0 = (integer) FAIL</p>

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

函式描述	打印文字
参数定义	<p>PosX 类型 : Integer 文字的 X 坐标位置(左上方水平位置), 单位为 dot</p> <p>PosY 类型 : Integer 文字的 Y 坐标位置(左上方垂直位置), 单位为 dot</p> <p>Height 类型 : Integer 字高</p> <p>FontName 类型 : (char *)</p>

	字体的名称 Data 类型 : (char *) 字符串数据内容
回传值	1 = (integer) OK 0 = (integer) FAIL

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

函式描述	打印文字 (Unicode 文字)
参数定义	PosX 类型 : Integer 文字的 X 坐标位置(左上方水平位置), 单位为 dot PosY 类型 : Integer 文字的 Y 坐标位置(左上方垂直位置), 单位为 dot Height 类型 : Integer 字高 FontName 类型 : (char *) 字体的名称 Data 类型 : (char *) Byte 数据内容 Len 类型 : Integer 数据长度
回传值	1 = (integer) OK 0 = (integer) FAIL

3.22 **ecTextOutR**(PosX, PosY, Height, FontName, Data, Width, Weight , Degree)

函式描述	打印文字 (可设定旋转)
参数定义	<p>PosX 类型 : Integer 文字的 X 坐标位置(左上方水平位置), 单位为 dot</p> <p>PosY 类型 : Integer 文字的 Y 坐标位置(左上方垂直位置), 单位为 dot</p> <p>Height 类型 : Integer 字高</p> <p>FontName 类型 : (char *) 字体的名称</p> <p>Data 类型 : (char *) 字符串数据内容</p> <p>Width 类型 : Integer 字宽</p> <p>Weight 类型 : Integer 字体粗细, 可设定范围为 0 ~ 1000 400 为一般文字 700 为粗体字</p> <p>Degree 类型 : Integer 旋转角度 0 : degree 0, 90 : degree 90 180 : degree 180 270 : degree 270</p>
回传值	<p>1 = (integer) OK 0 = (integer) FAIL</p>

3.23 **ecTextOutRW**(PosX, PosY, Height, FontName, Data, Width, Weight, Deg, Len)

函式描述	打印文字 (Unicode 文字，可设定旋转)
参数定义	<p>PosX 类型 : Integer 文字的 X 坐标位置(左上方水平位置)，单位为 dot</p> <p>PosY 类型 : Integer 文字的 Y 坐标位置(左上方垂直位置)，单位为 dot</p> <p>Height 类型 : Integer 字高</p> <p>FontName 类型 : (char *) 字体的名称</p> <p>Data 类型 : (char *) Byte 数据内容</p> <p>Width 类型 : Integer 字宽</p> <p>Weight 类型 : Integer 字体粗细，可设定范围为 0 ~ 1000 400 为一般文字 700 为粗体字</p> <p>Deg 类型 : Integer 旋转角度 0 : degree 0, 90 : degree 90 180 : degree 180 270 : degree 270</p> <p>Len 类型 : Integer 数据长度</p>
回传值	<p>1 = (integer) OK</p> <p>0 = (integer) FAIL</p>

3.24 ecTextOutFine(PosX, PosY, Height, FontName, Data, Width, Weight, Degree, Italic, Underline, Strikeout, Inverse)

函数描述	打印文字 (可设定旋转, 黑度, 斜体, 底线, 删除线, 颜色反向)
参数定义	<p>PosX 类型 : Integer 文字的 X 坐标位置(左上方水平位置), 单位为 dot</p> <p>PosY 类型 : Integer 文字的 Y 坐标位置(左上方垂直位置), 单位为 dot</p> <p>Height 类型 : Integer 字高</p> <p>FontName 类型 : (char *) 字体的名称</p> <p>Data 类型 : (char *) 字符串数据内容</p> <p>Width 类型 : Integer 字宽</p> <p>Weight 类型 : Integer 字体粗细, 可设定范围为 0 ~ 1000 400 为一般文字 700 为粗体字</p> <p>Degree 类型 : Integer 旋转角度 0 : degree 0, 90 : degree 90 180 : degree 180 270 : degree 270</p> <p>Italic 类型 : Integer 设定 TRUE(1) 则为斜体字, FALSE(0) 为一般字</p> <p>Underline</p>

	<p>类型 : Integer</p> <p>设定 TRUE(1) 则为底线字, FALSE(0) 为一般字</p> <p>Strikeout</p> <p>类型 : Integer</p> <p>设定 TRUE(1) 则为删除线字, FALSE(0) 为一般字</p> <p>Inverse</p> <p>类型 : Integer</p> <p>设定 TRUE(1) 则颜色反向显示, FALSE(0) 为正常显示</p>
回传值	<p>1 = (integer) OK</p> <p>0 = (integer) FAIL</p>

3.25 **ecTextOutFineW**(PosX, PosY, Height, FontName, Data, Width, Weight, Degree, Italic, Underline, Strikeout, Inverse, Len)

函式描述	打印文字 (Unicode 文字, 可设定旋转, 黑度, 斜体, 底线, 删除线, 颜色反向)
参数定义	<p>PosX</p> <p>类型 : Integer</p> <p>文字的 X 坐标位置(左上方水平位置), 单位为 dot</p> <p>PosY</p> <p>类型 : Integer</p> <p>文字的 Y 坐标位置(左上方垂直位置), 单位为 dot</p> <p>Height</p> <p>类型 : Integer</p> <p>字高</p> <p>FontName</p> <p>类型 : (char *)</p> <p>字体的名称</p> <p>Data</p> <p>类型 : (char *)</p> <p>字符串数据内容</p> <p>Width</p> <p>类型 : Integer</p> <p>字宽</p> <p>Weight</p> <p>类型 : Integer</p> <p>字体粗细, 可设定范围为 0 ~ 1000</p> <p>400 为一般文字</p>

Fabulous FBio DLL User Manual

	<p>700 为粗体字</p> <p>Degree</p> <p>类型 : Integer</p> <p>旋转角度</p> <p>0 : degree 0,</p> <p>90 : degree 90</p> <p>180 : degree 180</p> <p>270 : degree 270</p> <p>Italic</p> <p>类型 : Integer</p> <p>设定 TRUE(1) 则为斜体字, FALSE(0) 为一般字</p> <p>Underline</p> <p>类型 : Integer</p> <p>设定 TRUE(1) 则为底线字, FALSE(0) 为一般字</p> <p>Strikeout</p> <p>类型 : Integer</p> <p>设定 TRUE(1) 则为删除线字, FALSE(0) 为一般字</p> <p>Inverse</p> <p>类型 : Integer</p> <p>设定 TRUE(1) 则颜色反向显示, FALSE(0) 为正常显示</p> <p>Len</p> <p>类型 : Integer</p> <p>数据长度</p>
回传值	<p>1 = (integer) OK</p> <p>0 = (integer) FAIL</p>

3.26 **ecTextDownload**(Height, FontName, Data , Width, Weight , Deg, ID_Name)

函式描述	将文字内容转成一个图文件，下载至打印机中
参数定义	<p>Height 类型 : Integer 字高</p> <p>FontName 类型 : (char *) 字体的名称</p> <p>Data 类型 : (char *) 字符串数据内容</p> <p>Width 类型 : Integer 字宽</p> <p>Weight 类型 : Integer 字体粗细，可设定范围为 0 ~ 1000 400 为一般文字 700 为粗体字</p> <p>Deg 类型 : Integer 旋转角度 0 : degree 0, 90 : degree 90 180 : degree 180 270 : degree 270</p> <p>ID_Name 类型 : (char *) 上传后打印图文件指令(^Y)要呼叫的名称</p>
回传值	1 = (integer) OK 0 = (integer) FAIL

3.27 **ecTextDownloadW**(Height, FontName, Data , Width, Weight , Degree, ID_Name, Len)

函式描述	将文字(Unicode 文字)内容转成一个图文件，下载至打印机中
参数定义	<p>Height 类型 : Integer 字高</p> <p>FontName 类型 : (char *) 字体的名称</p> <p>Data 类型 : (char *) Byte 数据内容</p> <p>Width 类型 : Integer 字宽</p> <p>Weight 类型 : Integer 字体粗细，可设定范围为 0 ~ 1000 400 为一般文字 700 为粗体字</p> <p>Degree 类型 : Integer 旋转角度 0 : degree 0, 90 : degree 90 180 : degree 180 270 : degree 270</p> <p>ID_Name 类型 : (char *) 上传后打印图文件指令(^Y)要呼叫的名称</p> <p>Len 类型 : Integer 数据长度</p>
回传值	<p>1 = (integer) OK</p> <p>0 = (integer) FAIL</p>

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

函式描述	打印 1D 条形码
参数定义	<p>CodeType 类型 : (char *) 条形码类型 (请参考附件一)</p> <p>PosX 类型 : Integer 条形码的 X 坐标位置(左上方水平位置), 单位为 dot</p> <p>PosY 类型 : Integer 条形码的 Y 坐标位置(左上方垂直位置), 单位为 dot</p> <p>Narrow 类型 : Integer 条形码细线宽, 单位为 dot, 一般设定范围为 1 ~ 10 ITF14 (DUN14) 的设定范围为 5 ~ 8 UPC/EAN 系列条形码的设定范围为 2 ~ 4</p> <p>Wide 类型 : Integer 条形码粗线宽, 单位为 dot, 设定范围为 2 ~ 30 此参数仅 Code 39, 93, CODABAR & I 2 of 5 有作用</p> <p>Height 类型 : Integer 条形码高度, 单位为 dot, 设定范围为 24 ~ 1200</p> <p>Rotation 类型 : Integer 条形码旋转方式 0) 0°, 1) 90°, 2) 180°, 3) 270°</p> <p>Readable 类型 : Integer 码文显示方式 0: 不显示 1: 条形码下方置左 2: 条形码上方置左 3: 条形码下方置中 4: 条形码上方置中 5: 条形码下方置右 6: 条形码上方置右</p>

Fabulous FBio DLL User Manual

	Data 类型 : (char *) 条形码数据内容
回传值	1 = (integer) OK 0 = (integer) FAIL

3.29 Bar_S(CodeType, PosX, PosY, Data)

函式描述	打印 1D 条形码 (简易模式)
参数定义	CodeType 类型 : (char *) 条形码类型 PosX 类型 : Integer 条形码的 X 坐标位置(左上方水平位置), 单位为 dot PosY 类型 : Integer 条形码的 Y 坐标位置(左上方垂直位置), 单位为 dot Data 类型 : (char *) 条形码数据内容
回传值	1 = (integer) OK 0 = (integer) FAIL

3.30 Bar_GS1DataBar(CodeType, PosX, PosY, Narrow, Segment, Height, Rotation, Readable, Data)

函数描述	打印 GS1 DataBar 条形码
参数定义	<p>CodeType 类型 : (char *) 条形码类型 “0” : GS1 Databar Omnidirectional “1” : GS1 Databar Truncated “2” : GS1 Databar Stacked “3” : GS1 Databar Stacked Omnidir. “4” : GS1 Databar Limited “5” : GS1 Databar Expanded “6” : GS1 Databar Expanded Stacked</p> <p>PosX 类型 : Integer 条形码的 X 坐标位置(左上方水平位置), 单位为 dot</p> <p>PosY 类型 : Integer 条形码的 Y 坐标位置(左上方垂直位置), 单位为 dot</p> <p>Narrow 类型 : Integer 条形码细线宽, 单位为 dot, 一般设定范围为 1 ~ 10 ITF14 (DUN14) 的设定范围为 5 ~ 8 UPC/EAN 系列条形码的设定范围为 2 ~ 4</p> <p>Segment 类型 : Integer 设定每行条形码的最大宽度, 设定范围为 2~22, 此参数只适用于"GS1 Databar Expanded Stacked"</p> <p>Height 类型 : Integer 保留参数, 请固定设为 0</p> <p>Rotation 类型 : Integer 条形码旋转方式 0) 0°, 1) 90°, 2) 180°, 3) 270°</p> <p>Readable 类型 : Integer</p>

	码文显示方式 0: 不显示 1: 条形码下方置左 Data 类型 : (char *) 条形码数据内容
回传值	1 = (integer) OK 0 = (integer) FAIL

3.31 Bar_GS1DataBar_S(CodeType, PosX, PosY, Data)

函式描述	打印 GS1 DataBar 条形码 (简易模式)
参数定义	CodeType 类型 : (char *) 条形码类型 “0” : GS1 Databar Omnidirectional “1” : GS1 Databar Truncated “2” : GS1 Databar Stacked “3” : GS1 Databar Stacked Omnidir. “4” : GS1 Databar Limited “5” : GS1 Databar Expanded “6” : GS1 Databar Expanded Stacked PosX 类型 : Integer 条形码的 X 坐标位置(左上方水平位置), 单位为 dot PosY 类型 : Integer 条形码的 Y 坐标位置(左上方垂直位置), 单位为 dot Data 类型 : (char *) 条形码数据内容
回传值	1 = (integer) OK 0 = (integer) FAIL

3.32 Bar_PDF417 (PosX, PosY, Width, Height, Row, Col, ErrLevel, Len, Deg, Data)

函式描述	打印 PDF 417 条形码
参数定义	<p>PosX 类型 : Integer 条形码的 X 坐标位置(左上方水平位置), 单位为 dot</p> <p>PosY 类型 : Integer 条形码的 Y 坐标位置(左上方垂直位置), 单位为 dot</p> <p>Width 类型 : Integer 条形码或空白处, 最窄部份之宽度</p> <p>Height 类型 : Integer 条形码高度</p> <p>Row 类型 : Integer 条形码行数, 范围为 3 ~ 90, 若设定为 0 则会自动计算</p> <p>Col 类型 : Integer 条形码列数, 范围为 1 ~ 30. 若设定为 0 则会自动计算</p> <p>ErrLevel 类型 : Integer 错误修正度, 范围为 0 ~ 8</p> <p>Len 类型 : Integer 数据长度 (换行符号也需计算)</p> <p>Deg 类型 : Integer 条形码旋转方式 0) 0°, 1) 90°, 2) 180°, 3) 270°</p> <p>Data 类型 : (char *) 条形码数据内容</p>
回传值	<p>1 = (integer) OK</p> <p>0 = (integer) FAIL</p>

3.33 Bar_PDF417_S (PosX, PosY, Len, Data)

函式描述	打印 PDF 417 条形码 (简易模式)
参数定义	<p>PosX 类型 : Integer 条形码的 X 坐标位置(左上方水平位置), 单位为 dot</p> <p>PosY 类型 : Integer 条形码的 Y 坐标位置(左上方垂直位置), 单位为 dot</p> <p>Len 类型 : Integer 数据长度 (换行符号也需计算)</p> <p>Data 类型 : (char *) 条形码数据内容</p>
回传值	<p>1 = (integer) OK</p> <p>0 = (integer) FAIL</p>

3.34 Bar_MicroPDF417 (PosX, PosY, Width, Height, Mode, Len, Degree, Data)

函式描述	打印 Micro PDF 417 条形码
参数定义	<p>PosX 类型 : Integer 条形码的 X 坐标位置(左上方水平位置), 单位为 dot</p> <p>PosY 类型 : Integer 条形码的 Y 坐标位置(左上方垂直位置), 单位为 dot</p> <p>Width 类型 : Integer 条形码或空白处, 最窄部份之宽度</p> <p>Height 类型 : Integer 条形码高度</p> <p>Mode 类型 : Integer 数据模式, 范围为 0 ~ 33</p> <p>Len 类型 : Integer 数据长度 (换行符号也需计算)</p> <p>Degree 类型 : Integer 条形码旋转方式 0) 0°, 1) 90°, 2) 180°, 3) 270°</p> <p>Data 类型 : (char *) 条形码数据内容</p>
回传值	<p>1 = (integer) OK</p> <p>0 = (integer) FAIL</p>

3.35 Bar_MicroPDF417_S (PosX, PosY, Len, Data)

函式描述	打印 Micro PDF 417 条形码 (简易模式)
参数定义	<p>PosX 类型 : Integer 条形码的 X 坐标位置(左上方水平位置), 单位为 dot</p> <p>PosY 类型 : Integer 条形码的 Y 坐标位置(左上方垂直位置), 单位为 dot</p> <p>Len 类型 : Integer 数据长度 (换行符号也需计算)</p> <p>Data 类型 : (char *) 条形码数据内容</p>
回传值	<p>1 = (integer) OK</p> <p>0 = (integer) FAIL</p>

3.36 Bar_Maxicode (PosX, PosY, SymbolNo, SetNo, Mode, CountryCode, PostalCode, Class, Degree, Data)

函数描述	打印 Maxicode 条形码
参数定义	<p>PosX 类型 : Integer 条形码的 X 坐标位置(左上方水平位置), 单位为 dot</p> <p>PosY 类型 : Integer 条形码的 Y 坐标位置(左上方垂直位置), 单位为 dot</p> <p>SymbolNo 类型 : Integer 符号数, 设定范围为: 1 ~ 8.</p> <p>SetNo 类型 : Integer 符号组数, 设定范围为: 1 ~ 8 组</p> <p>Mode 类型 : Integer 数据模式, 可设定值为 2, 3, 4, 6 2: numeric postal code for U.S. 3: numeric postal code for non-U.S. 4: standard symbol, secretary 6: reader program, secretary</p> <p>CountryCode 类型 : (char *) 3 位数的国家码</p> <p>PostalCode 类型 : (char *) 邮政区号 美洲地区为 9 位数, 若仅有 5 位, 其余 4 位必须补 0 美洲以外地区应为 6 位数(文字数字混合)</p> <p>Class 类型 : (char *) 3 位数的服务等级</p> <p>Degree 类型 : Integer 条形码旋转方式 0) 0°, 1) 90°, 2) 180°, 3) 270°</p>

	Data 类型 : (char *) 条形码数据内容, 内容限制为 1 ~ 84 个字符
回传值	1 = (integer) OK 0 = (integer) FAIL

3.37 Bar_Maxicode_S (PosX, PosY, CountryCode, PostalCode, Class, Degree, Data)

函式描述	打印 Maxicode 条形码 (简易模式)
参数定义	PosX 类型 : Integer 条形码的 X 坐标位置(左上方水平位置), 单位为 dot PosY 类型 : Integer 条形码的 Y 坐标位置(左上方垂直位置), 单位为 dot CountryCode 类型 : (char *) 3 位数的国家码 PostalCode 类型 : (char *) 邮政区号 美洲地区为 9 位数, 若仅有 5 位, 其余 4 位必须补 0 美洲以外地区应为 6 位数(文字数字混合) Class 类型 : (char *) 3 位数的服务等级 Degree 类型 : Integer 条形码旋转方式 0) 0°, 1) 90°, 2) 180°, 3) 270° Data 类型 : (char *) 条形码数据内容, 内容限制为 1 ~ 84 个字符
回传值	1 = (integer) OK 0 = (integer) FAIL

3.38 Bar_DataMatrix (PosX, PosY, Enlarge, RotationR, Len, Data)

函式描述	打印 Data Matrix Code 条形码
参数定义	<p>PosX 类型 : Integer 条形码的 X 坐标位置(左上方水平位置), 单位为 dot</p> <p>PosY 类型 : Integer 条形码的 Y 坐标位置(左上方垂直位置), 单位为 dot</p> <p>Enlarge 类型 : Integer 条形码放大倍数(垂直及水平), 可设定范围为 1 ~ 40</p> <p>RotationR 类型 : (char *) 条形码旋转方式 “0” : 0° 的正方形条形码 “1” : 90° 的正方形条形码 “2” : 180° 的正方形条形码 “3” : 270° 的正方形条形码 “0R”: 0° 的矩形条形码 “1R”: 90° 的矩形条形码 “2R”: 180° 的矩形条形码 “3R”: 270° 的矩形条形码</p> <p>Len 类型 : Integer 数据长度</p> <p>Data 类型 : (char *) 条形码数据内容</p>
回传值	1 = (integer) OK 0 = (integer) FAIL

3.39 Bar_DataMatrix_S (PosX, PosY, Len, Data)

函式描述	打印 Data Matrix Code 条形码 (简易模式)
参数定义	<p>PosX 类型 : Integer 条形码的 X 坐标位置(左上方水平位置), 单位为 dot</p> <p>PosY 类型 : Integer 条形码的 Y 坐标位置(左上方垂直位置), 单位为 dot</p> <p>Len 类型 : Integer 数据长度</p> <p>Data 类型 : (char *) 条形码数据内容</p>
回传值	<p>1 = (integer) OK</p> <p>0 = (integer) FAIL</p>

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

函数描述	打印 QR Code 条形码
参数定义	<p>PosX 类型 : Integer 条形码的 X 坐标位置(左上方水平位置), 单位为 dot</p> <p>PosY 类型 : Integer 条形码的 Y 坐标位置(左上方垂直位置), 单位为 dot</p> <p>Mode 类型 : Integer 编码模式 1: Numerical data mode 2: Alpha numerical data mode 3: 8-bit data mode 4: Kanji data mode 5: Mixing mode (不适用于 Micro QR Code)</p> <p>Type 类型 : Integer 设定类型 1: Model1 (original) 2: Model2 (enhanced) 3: Micro QR Code</p> <p>ErrLevel 类型 : (char *) 错误修正等级 L: Low M: Medium Q: Medium high H: High (不适用于 Micro QR Code)</p> <p>Mask 类型 : Integer 屏蔽因子, 设定范围为 0~7 或是设定为 8 自动切换 若为 Micro QR Code 时, 此值必须设为 0</p> <p>Mul 类型 : Integer 放大倍数, 设定范围为 1 ~ 40</p> <p>Len</p>

	类型 : Integer 数据长度 Deg 类型 : Integer 条形码旋转方式 0) 0°, 1) 90°, 2) 180°, 3) 270° Data 类型 : (char *) 条形码数据内容
回传值	1 = (integer) OK 0 = (integer) FAIL

3.41 Bar_QRcode_S (PosX, PosY, Len, Data)

函式描述	打印 QR Code 条形码 (简易模式)
参数定义	PosX 类型 : Integer 条形码的 X 坐标位置(左上方水平位置), 单位为 dot PosY 类型 : Integer 条形码的 Y 坐标位置(左上方垂直位置), 单位为 dot Len 类型 : Integer 数据长度 Data 类型 : (char *) 条形码数据内容
回传值	1 = (integer) OK 0 = (integer) FAIL

3.42 Bar_Aztec (PosX, PosY, Degree, Mul, ECICs, Type, MenuSymbol, Len, Data)

函数描述	打印 Aztec 条形码
参数定义	<p>PosX 类型 : Integer 条形码的 X 坐标位置(左上方水平位置), 单位为 dot</p> <p>PosY 类型 : Integer 条形码的 Y 坐标位置(左上方垂直位置), 单位为 dot</p> <p>Degree 类型 : Integer 条形码旋转方式 0) 0°, 1) 90°, 2) 180°, 3) 270°</p> <p>Mul 类型 : Integer 分辨率 1: 150 dpi printers 2: 200 dpi printers 3: 300 dpi printers 6: 600 dpi printers</p> <p>ECICs 类型 : (char *) 是否包含 ECICs (Extended Channel Interpretation Code) “Y”: 包含 ECIC 资料 “N”: 不包含 ECIC 资料</p> <p>Type 类型 : Integer 0 : 默认的错误修正等级 01 ~ 99 : 最小错误修正率 (0%~99%) 101 ~ 104: Compact Symbol 1 ~4 Layer 201 ~ 232: Full Symbol 1 ~ 31 Layer 300 : Rune Symbol</p> <p>MenuSymbol 类型 : (char *) 是否为 Menu Sysbol (条形码扫描仪的初始化符号) “Y”: 是 Menu Sysbol “N”: 不是 Menu Sysbol</p>

	Len 类型 : Integer 数据长度 Data 类型 : (char *) 条形码数据内容
回传值	1 = (integer) OK 0 = (integer) FAIL

3.43 Bar_Aztec_S (PosX, PosY, Len, Data)

函式描述	打印 Aztec 条形码 (简易模式)
参数定义	PosX 类型 : Integer 条形码的 X 坐标位置(左上方水平位置), 单位为 dot PosY 类型 : Integer 条形码的 Y 坐标位置(左上方垂直位置), 单位为 dot Len 类型 : Integer 数据长度 Data 类型 : (char *) 条形码数据内容
回传值	1 = (integer) OK 0 = (integer) FAIL

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

函式描述	打印内建字或亚洲字 (打印机内必须已经先下载亚洲字)
参数定义	<p>FontType Type : (char *) 字体型态 (请参考附件二)</p> <p>PosX Type : Integer 文字的 X 坐标位置(左上方水平位置), 单位为 dot</p> <p>PosY Type : Integer 文字的 Y 坐标位置(左上方垂直位置), 单位为 dot</p> <p>Mul_X Type : Integer 水平放大倍数 (最多 8 倍)</p> <p>Mul_Y Type : Integer 垂直放大倍数 (最多 8 倍)</p> <p>Gap Type : Integer 字符间的空白间距, 可设定范围为 0 ~ 200 dot</p> <p>RotationInverse Type : (char *) 内建字设定值范围为 0 ~ 3, 亚洲字设定范围为 0 ~ 7 0) 0°, 1) 90°, 2) 180°, 3) 270° 4) 0°, 5) 90°, 6) 180°, 7) 270° 0~3 为字符串中所有字符一起旋转 4~7 为字符分别旋转 在此参数最后加上字符 "I" 时, 可以黑白反向打印</p> <p>Data Type : (char *) 字符串数据</p>
回传值	<p>1 = (integer) OK</p> <p>0 = (integer) FAIL</p>

3.45 **InternalFont_TextOut_S** (FontType, PosX, PosY, Data)

函式描述	打印内建字或亚洲字 (简易模式)
参数定义	<p>FontType Type : (char *) 字体形态 (请参考附件二)</p> <p>PosX Type : Integer 文字的 X 坐标位置(左上方水平位置), 单位为 dot</p> <p>PosY Type : Integer 文字的 Y 坐标位置(左上方垂直位置), 单位为 dot</p> <p>Data Type : (char *) 字符串数据</p>
回传值	<p>1 = (integer) OK</p> <p>0 = (integer) FAIL</p>

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

函式描述	打印下载的英数字 (打印机内必须已经先下载英数字)
参数定义	<p>FontName Type : (char *) 字体代称. 可设定范围为 A ~ Z (名称必须对应到已下载的字体)</p> <p>PosX Type : Integer 文字的 X 坐标位置(左上方水平位置), 单位为 dot</p> <p>PosY Type : Integer 文字的 Y 坐标位置(左上方垂直位置), 单位为 dot</p> <p>Mul_X Type : Integer 水平放大倍数 (最多 8 倍)</p> <p>Mul_Y Type : Integer 垂直放大倍数 (最多 8 倍)</p> <p>Gap Type : Integer 字符间的空白间距, 可设定范围为 0 ~ 200 dot</p> <p>RotationInverse Type : (char *) 内建字设定值范围为 0 ~ 3, 亚洲字设定范围为 0 ~ 7 0) 0°, 1) 90°, 2) 180°, 3) 270° 4) 0°, 5) 90°, 6) 180°, 7) 270° 0~3 为字符串中所有字符一起旋转 4~7 为字符分别旋转 在此参数最后加上字符 "I" 时, 可以黑白反向打印</p> <p>Data Type : (char *) 字符串数据</p>
回传值	<p>1 = (integer) OK</p> <p>0 = (integer) FAIL</p>

3.47 DownloadFont_TextOut_S (FontName, PosX, PosY, Data)

函式描述	打印下载的英数字 (简易模式)
参数定义	<p>FontName Type : (char *) 字体代称. 可设定范围为 A ~ Z (名称必须对应到已下载的字体)</p> <p>PosX Type : Integer 文字的 X 坐标位置(左上方水平位置), 单位为 dot</p> <p>PosY Type : Integer 文字的 Y 坐标位置(左上方垂直位置), 单位为 dot</p> <p>Data Type : (char *) 字符串数据</p>
回传值	<p>1 = (integer) OK</p> <p>0 = (integer) FAIL</p>

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

函式描述	打印下载的向量字 (打印机内必须已经先下载向量字)
参数定义	<p>FontName Type : (char *) 字体代称. 可设定范围为 A ~ Z (名称必须对应到已下载的字体)</p> <p>PosX Type : Integer 文字的 X 坐标位置(左上方水平位置), 单位为 dot</p> <p>PosY Type : Integer 文字的 Y 坐标位置(左上方垂直位置), 单位为 dot</p> <p>Font_W Type : Integer 字体宽度缩放的倍率, 可设定范围为 8 ~ 2000 dot</p> <p>Font_H Type : Integer 字体高度缩放的倍率, 可设定范围为 8 ~ 2000 dot</p> <p>SpaceChar Type : Integer 字符间的空白间距, 可设定范围为 0 ~ 200 dot</p> <p>RotationInverse Type : (char *) 内建字设定值范围为 0 ~ 3, 亚洲字设定范围为 0 ~ 7 0) 0°, 1) 90°, 2) 180°, 3) 270° 4) 0°, 5) 90°, 6) 180°, 7) 270° 0~3 为字符串中所有字符一起旋转 4~7 为字符分别旋转 在此参数最后加上字符 "I" 时, 可以黑白反向打印</p> <p>TTFTable Type : (char *) 设定向量字查找的对应表(若设定错误会印出错误的文字) 若设定为 0: 使用内建的 ASCII 对应表 若设定为 A~Z: 使用使用者下载的 Unicode 对应表</p> <p>WidthMode Type : Integer</p>

	0：宽/高比模式 1：等宽模式 (暂时未开放) Data Type : (char *) 字符串数据
回传值	1 = (integer) OK 0 = (integer) FAIL

3.49 TrueTypeFont_TextOut_S (FontName, PosX, PosY, Data)

函式描述	打印下载的向量字 (简易模式)
参数定义	FontName Type : (char *) 字体代称. 可设定范围为 A ~ Z (名称必须对应到已下载的字体) PosX Type : Integer 文字的 X 坐标位置(左上方水平位置), 单位为 dot PosY Type : Integer 文字的 Y 坐标位置(左上方垂直位置), 单位为 dot Data Type : (char *) 字符串数据
回传值	1 = (integer) OK 0 = (integer) FAIL

3.50 DrawHorLine (PosX, PosY, Length, Thick)

函式描述	打印水平线
参数定义	<p>PosX Type : Integer 水平线左上角起点的 X 坐标, 单位为 dot</p> <p>PosY Type : Integer 水平线左上角起点的 Y 坐标, 单位为 dot</p> <p>Length Type : Integer 水平线长度, 单位为 dot</p> <p>Thick Type : Integer 水平线的线宽(粗细设定), 单位为 dot</p>
回传值	<p>1 = (integer) OK</p> <p>0 = (integer) FAIL</p>

3.51 DrawVerLine (PosX, PosY, Length, Thick)

函式描述	打印垂直线
参数定义	<p>PosX Type : Integer 垂直线左上角起点的 X 坐标, 单位为 dot</p> <p>PosY Type : Integer 垂直线左上角起点的 Y 坐标, 单位为 dot</p> <p>Length Type : Integer 垂直线长度, 单位为 dot</p> <p>Thick Type : Integer 垂直线的线宽(粗细设定), 单位为 dot</p>
回传值	<p>1 = (integer) OK</p> <p>0 = (integer) FAIL</p>

3.52 FillRec (PosX, PosY, Rec_W, Rec_H)

函数描述	打印实心矩形
参数定义	<p>PosX Type : Integer 矩形左上角的 X 坐标, 单位为 dot</p> <p>PosY Type : Integer 矩形左上角的 Y 坐标, 单位为 dot</p> <p>Rec_W Type : Integer 矩形宽度 (unit: dot)</p> <p>Rec_H Type : Integer 矩形高度 (unit: dot)</p>
回传值	<p>1 = (integer) OK</p> <p>0 = (integer) FAIL</p>

3.53 DrawRec (PosX, PosY, Rec_W, Rec_H, lrw, ubw)

函数描述	打印空心矩形
参数定义	<p>PosX Type : Integer 矩形左上角的 X 坐标, 单位为 dot</p> <p>PosY Type : Integer 矩形左上角的 Y 坐标, 单位为 dot</p> <p>Rec_W Type : Integer 矩形宽度 (unit: dot)</p> <p>Rec_H Type : Integer 矩形高度 (unit: dot)</p> <p>lrw Type : Integer 左右边缘的线宽(粗细设定), 单位为 dot</p> <p>ubw Type : Integer 上下边缘的线宽(粗细设定), 单位为 dot</p>

回传值	1 = (integer) OK 0 = (integer) FAIL
-----	--

3.54 DrawOblique (PosX1, PosY1, Thick, PosX2, PosY2)

函数描述	打印斜线
参数定义	<p>PosX1 Type : Integer 斜线起点的 X 坐标, 单位为 dot</p> <p>PosY1 Type : Integer 斜线起点的 Y 坐标, 单位为 dot</p> <p>Thick Type : Integer 线宽(粗细设定), 单位为 dot</p> <p>PosX2 Type : Integer 斜线终点的 X 坐标, 单位为 dot</p> <p>PosY2 Type : Integer 斜线终点的 Y 坐标, 单位为 dot</p>
回传值	1 = (integer) OK 0 = (integer) FAIL

3.55 DrawEllipse (PosX, PosY, Ellipse_W, Ellipse_H, Thick)

函数描述	Draw Ellipse
参数定义	<p>PosX Type : Integer 椭圆外接矩形的左上角 X 坐标, 单位为 dot</p> <p>PosY Type : Integer 椭圆外接矩形的左上角 Y 坐标, 单位为 dot</p> <p>Ellipse_W Type : Integer 椭圆外接矩形的宽, 单位为 dot</p> <p>Ellipse_H Type : Integer 椭圆外接矩形的高, 单位为 dot</p>

	Thick Type : Integer 线宽(粗细设定), 单位为 dot
回传值	1 = (integer) OK 0 = (integer) FAIL

3.56 DrawRoundRec (PosX, PosY, Rec_W, Rec_H, Arc_W, Arc_H, Thick)

函式描述	打印圆角矩形
参数定义	<p>PosX Type : Integer 圆角矩形外接矩形的左上角 X 坐标, 单位为 dot</p> <p>PosY Type : Integer 圆角矩形外接矩形的左上角 Y 坐标, 单位为 dot</p> <p>Rec_W Type : Integer 圆角矩形外接矩形的宽, 单位为 dot</p> <p>Rec_H Type : Integer 圆角矩形外接矩形的高, 单位为 dot</p> <p>Arc_W Type : Integer 角落圆弧形的外接矩形的宽</p> <p>Arc_H Type : Integer 角落圆弧形的外接矩形的高</p> <p>Thick Type : Integer 线宽(粗细设定), 单位为 dot</p>
回传值	1 = (integer) OK 0 = (integer) FAIL

3.57 DrawTriangle (PosX1, PosY1, PosX2, PosY2, PosX3, PosY3, Thick)

函式描述	打印三角形
参数定义	<p>PosX1 Type : Integer 三角形第 1 个顶点的 X 坐标</p> <p>PosY1 Type : Integer 三角形第 1 个顶点的 Y 坐标</p> <p>PosX2 Type : Integer 三角形第 2 个顶点的 X 坐标</p> <p>PosY2 Type : Integer 三角形第 2 个顶点的 Y 坐标</p> <p>PosX3 Type : Integer 三角形第 3 个顶点的 X 坐标</p> <p>PosY3 Type : Integer 三角形第 3 个顶点的 Y 坐标</p> <p>Thick Type : Integer 线宽(粗细设定), 单位为 dot</p>
回传值	<p>1 = (integer) OK</p> <p>0 = (integer) FAIL</p>

3.58 DrawDiamond (PosX, PosY, Diamond_W, Diamond_H, Thick)

函式描述	打印菱形
参数定义	<p>PosX Type : Integer 菱形外接矩形的左上角 X 坐标, 单位为 dot</p> <p>PosY Type : Integer 菱形外接矩形的左上角 Y 坐标, 单位为 dot</p> <p>Diamond_W Type : Integer 菱形外接矩形的宽, 单位为 dot</p>

Fabulous FBio DLL User Manual

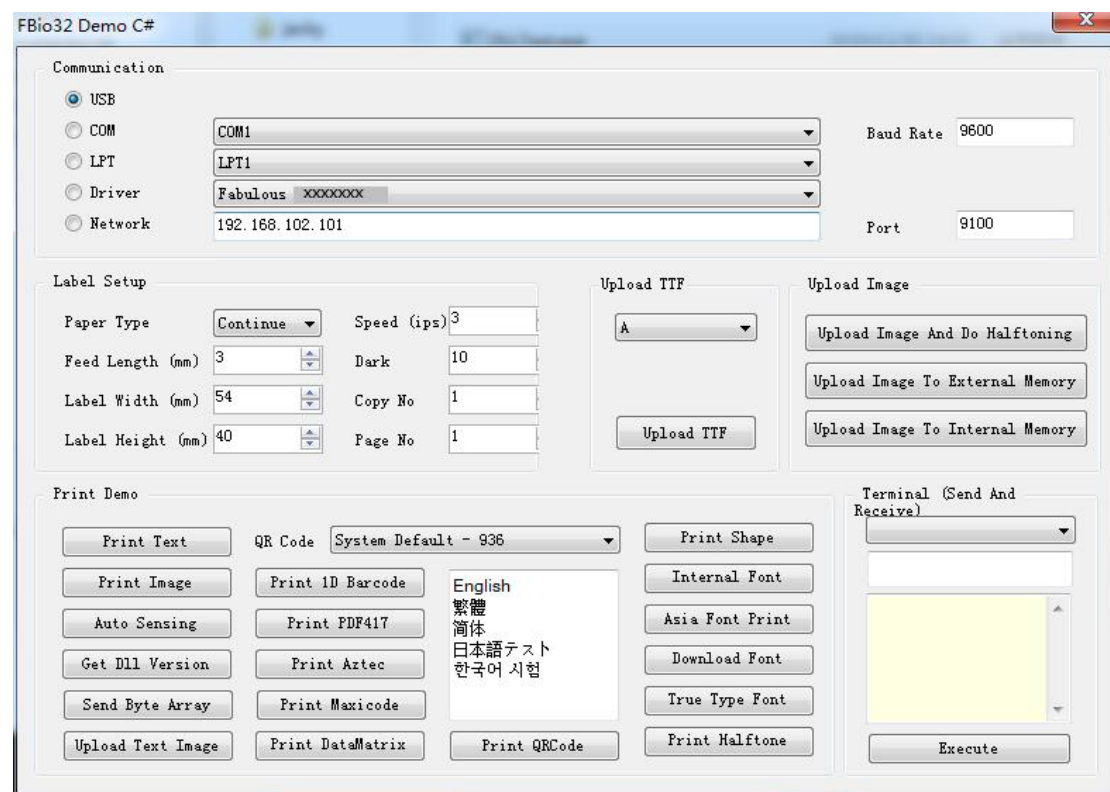
	<p>Diamond_H</p> <p>Type : Integer</p> <p>菱形外接矩形的高, 单位为 dot</p> <p>Thick</p> <p>Type : Integer</p> <p>线宽(粗细设定), 单位为 dot</p>
回传值	<p>1 = (integer) OK</p> <p>0 = (integer) FAIL</p>

3.59 GetDllVersion(Version)

函式描述	取得 EZio DLL 的版本资料
参数定义	<p>Version</p> <p>类型 : (char **)</p> <p>DLL 的版本数据</p>
回传值	数据长度

4. 程序开发指引

请参考 Example 目录中的范例程序代码。您可以在范例程序中，学会透过 USB、COM、LPT、网络及 Driver 通讯方式控制 GoDEX 打印机。同时也可以学会基本的文字、图文件与条形码的打印与打印机参数设定。



4.1 C# 程序开发

如果您想快速开发 C#程序，可以在[FBio_Sample_C#] 目录中取得 FBio32.dll、FBioApi.cs、FBPrinter.cs，并将这 3 个档案加入您的项目中。接着只要在您的项目中宣告以下对象

```
FBPrinter Printer = new FBPrinter();
```

就可以使用 Printer 对象快速的操作 Fabulous打印机。

4.2VB.NET 程序开发

如果您想快速开发 VB.NET 程序，可以在[FBio_Sample_VBNET] 目录中取得 FBio32.dll、FBioApi.vb、FBPrinter.vb，并将这 3 个档案加入您的项目中。接着只要在您的项目中宣告以下对象

```
Printer As FBPrinter = New FBPrinter()
```

就可以使用 Printer 对象快速的操作 Fabulous打印机。

4.3 BCB6 程序开发

如果您想快速开发 BCB6 程序，可以在[FBio_Sample_BCB6] 目录中取得 FBio32.dll、FBio32_BCB.lib、EZio32.h、ParaDef.h、clsPrinterCommand.cpp、clsPrinterCommand.h、clsPrinterConfig.cpp、clsPrinterConfig.h、FBPrinter.cpp、GodexPrinter.h，并将这些档案加入您的项目中。接着只要在您的项目中宣告以下对象

```
FBPrinter Printer;
```

就可以使用 Printer 对象快速的操作 Fabulous 打印机。

4.4 Delphi 程序开发

如果您想快速开发 Delphi 2010 程序，可以在[FBio_Sample_Delphi2010] 目录中取得 FBio32.dll、FBPrinter.pas，并将这 2 个档案加入您的项目中。接着只要在您的项目中宣告以下对象

```
mPrinter: TFBPrinter;
```

就可以使用 Printer 对象快速的操作 Fabulous 打印机。

4.5 VC (MFC) 程序开发

如果您想快速开发 VC 2008 程序，可以在[FBio_Sample_VC2008(MFC)] 目录中取得 FBio32.dll、FBio32_VC.lib、FBio32.h、ParaDef.h、clsPrinterCommand.cpp、clsPrinterCommand.h、clsPrinterConfig.cpp、clsPrinterConfig.h、FBPrinter.cpp、FBPrinter.h，并将这些档案加入您的项目中。接着只要在您的项目中宣告以下对象

```
FBPrinter Printer;
```

就可以使用 Printer 对象快速的操作 Fabulous打印机。

4.6 Java 程序开发

如果您想快速开发 Java 程序，可以在[FBio_Sample_Java_Luna] 目录中取得 FBio32.dll、jna-3.5.1.jar、FBPrinter.java、FBioLib.java、clsPrinterConfig.java、clsPrinterCommand.java，并将这些档案加入您的项目中。接着只要在您的项目中宣告以下对象

```
FBPrinter Printer = new FBPrinter();
```

就可以使用 Printer 对象快速的操作 Fabulous 打印机。

附件一 (一维条形码型态)

一维条形码的 Code Type 定义

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 check	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
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

附件二 (内建字型态)

[Internal Font](#)

内键字字体大小定义 (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		