
SDK 内容说明	4
1. PosAsynncTask.....	4
●构造器详细资料:	4
●方法详细资料:	4
doInBackground.....	4
onPostExecute	4
2. BackgroundInit	4
●方法详细资料:	4
doinbackground	5
3.IMyBinder	5
●方法详细资料:	5
connectNetPort	5
connectBtPort.....	5
connectUsbPort.....	6
disconnectCurrentPort.....	6
acceptdatafromprinter.....	6
readBuffer	6
clearBuffer	7
checkLinkedState.....	7
write.....	7
writeDataByYousef	7
4. ProcessData	7
●方法详细资料:	8
processDataBeforeSend.....	8
5. UiExecute.....	8
●方法详细资料:	8
onsuccess	8
onfailed	8
6. PosprinterService	9
7. BitmapToByteData.....	9
●枚举类型:	9
BmpType.....	9
●方法详细资料:	9
rasterBmpToSendData.....	9
flashBmpToSendData.....	10
downloadBmpToSendTSCData.....	10
downloadBmpToSendData.....	10
8. DataForSendToPrinterTSC	11
setCharsetName	11
sizeBymm	11
sizeByinch.....	12
sizeBydot	12
gapBymm.....	12
gapBydot.....	12

gapByinch	13
gapDetect	13
blinDetect	14
autoDetect	14
blinByinch	15
blinBymm.....	15
blinBydot.....	15
offsetByinch	16
offsetBymm.....	16
offsetBydot.....	16
speed.....	17
density	17
direction	18
reference	18
shift.....	18
country.....	19
codePage	19
cls	20
feed.....	20
backFeed	20
formFeed.....	21
home	21
print	21
sound	21
cut.....	22
limitFeedBymm	22
limitFeedByinch	22
limitFeedBydot	23
selfTest	23
ej.....	23
delay	24
disPlay	24
initialPrinter	24
bar	24
barCode.....	25
bitmap	25
box.....	26
circle	26
ellipse.....	27
codaBlockFMode.....	27
dmatrix.....	28
erase.....	28
pdf417.....	28
putBmp	29

putpcx	29
qrCode	30
reverse.....	30
text.....	31
block.....	31
checkPrinterStateByPort9100	32
checkPrinterStateByPort4000	32
9. DataForSendToPrinterPos58.....	33
10. DataForSendToPrinterPos76	33
11. DataForSendToPrinterPos80	33
12. PosPrinterDev.....	33
● 枚举类型:	34
portType.....	34
ErrorCode.....	34
● 构造器详细资料:	34
● 方法详细资料:	34
open	35
write.....	35
read.....	35
close.....	36
GetPortInfo.....	36
GetUsbPathNames	36
13. ReadExcel	36
● 方法详细资料:	37
readExcel	37
getExcelSheetName	37
14. RoundQueue.....	37
● 构造器详细资料:	37
● 方法详细资料:	38
addLast	38
removeFirst	38
realSize	38
isEmpty.....	38
isFull.....	39
clear	39
get.....	39
indexOf	40
gethead.....	40
gettail.....	40
getLast	40
getFirst	41

SDK 内容说明

1. PosAsyncncTask

说明：

异步任务类。

目录：

net.posprinter.asyncncTask.PosAsyncncTask

● 构造器详细资料：

```
public PosAsyncncTask(UiExecute execute, BackgroundInit init)
```

● 方法详细资料：

doInBackground

说明：

接口的抽象方法,具体实现在调用这个类的构造方法前实现。

函数构造：

```
protected boolean doInBackground(Void... params)
```

onPostExecute

说明：

接口的抽象方法,具体实现在调用这个类的构造方法前实现。

函数构造：

```
protected void onPostExecute(java.lang.Boolean result)
```

2. BackgroundInit

说明：

接口类，此接口的实现方法运行在异步任务的工作线程中，正常执行完成，返回 `true`，否则 `false`。

目录：

net.posprinter.posprinterinterface.BackgroundInit

● 方法详细资料：

doinbackground

说明:

无。

函数构造:

boolean doinbackground()

返回:

boolean 正常执行完成, 返回 true, 否则 false;

3.IMyBinder

说明:

接口类, 用于实现设备绑定。

目录:

net.posprinter.posprinterface.IMyBinder

●方法详细资料:

connectNetPort

说明:

连接网络端口。

函数构造:

void connectNetPort(String ethernetIP, int ethernetPort, UiExecute execute)

参数说明:

UiExecute - execute 接口的实现类, 实现方法在 UI 线程中执行;

ethernetIP 打印机 ip 地址;

ethernetPort 打印机端口号。

connectBtPort

说明:

连接蓝牙端口。

函数构造:

void connectBtPort(String bluetoothID, UiExecute execute)

参数说明:

UiExecute - execute 接口的实现类, 实现方法在 UI 线程中执行

bluetoothID 蓝牙名称。

connectUsbPort

说明：

连接 USB 端口。

函数构造：

```
void connectUsbPort(Context context, String usbID, UiExecute execute)
```

参数说明：

UiExecute - execute 接口的实现类，实现方法在 UI 线程中执行；

context 上下文；

usbID 蓝牙地址码。.

disconnectCurrentPort

说明：

断开当前端口,在不需连接打印机操作或退出应用时调用此方法，断开连接，释放内存，减轻系统负荷

函数构造：

```
void disconnectCurrentPort(UiExecute execute)
```

参数说明：

UiExecute - execute 接口的实现类，实现方法在 UI 线程中执行。

acceptdatafromprinter

说明：

开启一条线程来接收打印机发送的数据，并将数据存储在 size 为 500 的环形队列缓存区。

函数构造：

```
void acceptdatafromprinter(UiExecute execute)
```

参数说明：

data - 机接收的数据为 byte[]，字节数组。

readBuffer

说明：

读取环形缓存区的数据。

函数构造：

```
RoundQueue<byte[]> readBuffer()
```

返回：

返回一个环形队列对象。

clearBuffer

说明：

清除环形缓存区的数据。

函数构造：

`void clearBuffer()`

checkLinkState

说明：

检测 app 与打印机的连接状态。

函数构造：

`void checkLinkState(UiExecute execute)`

参数说明：

UiExecute - execute 接口的实现类，实现方法在 UI 线程中执行。

write

说明：

处理好的数据发送给打印机。

函数构造：

`void write(byte[] data, UiExecute execute)`

参数说明：

data - 打印机接收的数据为 byte[]，字节数组。

writeDataByYouself

说明：

可以自己实现 ProcessData 接口方法，来调用工具类的指令调配方法发送自定义的指令集合，适合在此接口实现耗时的数据处理操作。

函数构造：

`void writeDataByYouself(UiExecute execute, ProcessData processData)`

参数说明：

execute - execute 接口的实现类，实现方法在 UI 线程中执行；

processData 接口的实现类，可以重写要发送的数据的处理方法。

4. ProcessData

说明：

接口类，用于发送的数据。

目录：

net.posprinter.posprinterinterface.ProcessData

●方法详细资料：

processDataBeforeSend

说明：

此接口的实现方法中处理处理要发送的数据，切执行在异步任务的工作现场中，此方法适合处理复杂的，数量较大的数据。

函数构造：

List<byte[]> processDataBeforeSend();

返回：

List,一个形参为 byte[]数组的集合可将要发送的数据添加到这个集合里。

5. UiExecute

说明：

接口类，用于发送的数据。

目录：

net.posprinter.posprinterinterface.UiExecute

●方法详细资料：

onsuccess

说明：

数据发送成功后执行的 UI 操作

函数构造：

void onsuccess()

onfailed

说明：

数据发送失败后，在主线程中的操作

函数构造：

void onfailed()

6. PosprinterService

说明:

服务类, 继承 IMyBinder, 内部方法同 IMyBinder。

目录:

net.posprinter.service.PosprinterService

7. BitmapToByteData

说明:

图片处理类, 将 bitmap 去灰度并转化为 byte[]。

目录:

net.posprinter.utils.BitmapToByteData

● 枚举类型:

BmpType

说明:

图片处理方式。

变量:

Dithering: 抖动算法;

Threshold: 二值法

● 方法详细资料:

rasterBmpToSendData

说明:

将图片 bitmap 对象转换成可直接发送给打印机的 byte[], 也是这个类唯一可以调用的静态方法, 供光栅位图打印的方法

函数构造:

```
public static byte[] rasterBmpToSendData(int m, Bitmap mBitmap,
BitmapToByteData.BmpType bmpType)
```

参数:

mBitmap - Bitmap 对象
bmpType - 对图片处理的算法类型选择。
返回：
byte[]字节数组。

flashBmpToSendData

说明：
将图片 bitmap 对象转换成可直接发送给打印机的 byte[],也是这个类唯一可以调用的静态方法，供 Flash 位图下载提供的方法

函数构造：
public static byte[] flashBmpToSendData(int m, Bitmap mBitmap, BitmapToByteData.BmpType bmpType)

参数：
mBitmap - Bitmap 对象
bmpType - 对图片处理的算法类型选择。

返回：
byte[]字节数组。

downloadBmpToSendTSCData

说明：
将图片 bitmap 对象转换成可直接发送给打印机的 byte[],也是这个类唯一可以调用的静态方法，供 TSC 定义下载位图指令使用

函数构造：
public static byte[] downloadBmpToSendTSCData(int m, Bitmap mBitmap, BitmapToByteData.BmpType bmpType)

参数：
mBitmap - Bitmap 对象
bmpType - 对图片处理的算法类型选择。

返回：
byte[]字节数组。

downloadBmpToSendData

说明：
将图片 bitmap 对象转换成可直接发送给打印机的 byte[],也是这个类唯一可以调用的静态方法，供定义下载位图指令使用

函数构造：
public static byte[] downloadBmpToSendData (int m, Bitmap mBitmap, BitmapToByteData.BmpType bmpType)

参数:

mBitmap - Bitmap 对象
bmpType - 对图片处理的算法类型选择.

返回:

byte[] 字节数组。

8. DataForSendToPrinterTSC

说明:

指令处理工具类。该类处理的指令只适用于条码打印机系列的产品；具体指令的描述和效果请参考编程手册 TSC 条码打印机指令 TSPL 手册.pdf。

目录:

net.posprinter.utils.DataForSendToPrinterTSC

setCharsetName

说明:

设置将打印内容传输给打印机所采用的字符编码，默认为中文编码“gbk”

函数构造:

```
public static void setCharsetName(java.lang.String charset)
```

参数:

charset - 字符编码类型（打印机所能识别的字符编码类型，具体请参考编程手册代码页设置）

返回:

无

sizeBymm

说明:

用于设定标签纸的宽度及长度，单位为毫米（mm）。

函数构造:

```
public static byte[] sizeBymm(double m, double n)
```

参数:

m - 标签长度
n - 标签宽度

返回:

byte[]

sizeByinch

说明:

用于设定标签纸的宽度及长度，单位为英寸（inch）。

函数构造:

```
public static byte[] sizeByinch(double m, double n)
```

参数:

m - 标签长度

n - 标签宽度

返回:

byte[]

sizeBydot

说明:

用于设定标签纸的宽度及长度，单位为像素（mm）。

函数构造:

```
public static byte[] sizeBydot(int m, int n)
```

参数:

m - 标签长度

n - 标签宽度

返回:

byte[]

gapBymm

说明:

用于设定两张标签纸中间的垂直间距，单位为像素（mm）。

函数构造:

```
public static byte[] gapBymm(double m, double n)
```

参数:

m - 两张标签纸间的间隙高度

$0 \leq m \leq 1$ (inch), $0 \leq m \leq 25.4$ (mm) $0 \leq m \leq 5$ (inch),

$0 \leq m \leq 127$ (mm) / V6.21 及之后 firmware

n - 间隙偏移量

$n \leq \text{label length}$ (inch or mm)

返回:

byte[]

gapBydot

说明：

用于设定两张标签纸中间的垂直间距，单位为像素（dot）。

函数构造：

```
public static byte[] gapBydot(int m, int n)
```

参数：

m - 两张标签纸间的间隙高度

$0 \leq m \leq 1$ (inch), $0 \leq m \leq 25.4$ (mm) $0 \leq m \leq 5$ (inch),

$0 \leq m \leq 127$ (mm) / V6.21及之后 firmware

n - 间隙偏移量

$n \leq \text{label length}$ (inch or mm)

返回：

byte[]

gapByinch

说明：

用于设定两张标签纸中间的垂直间距，单位为英寸（inch）。

函数构造：

```
public static byte[] gapByinch(double m, double n)
```

参数：

m - 两张标签纸间的间隙高度

$0 \leq m \leq 1$ (inch), $0 \leq m \leq 25.4$ (mm) $0 \leq m \leq 5$ (inch),

$0 \leq m \leq 127$ (mm) / V6.21及之后 firmware

n - 间隙偏移量

$n \leq \text{label length}$ (inch or mm)

返回：

byte[]

gapDetect

说明：

此条指令会使打印机持续出纸，通过纸张及间隙通过间隙感测器时所侦测到的强度差异，来区分定义纸张及间隙的大小。一般状况下使用此条指令所侦测到的纸张大小及间隙大小可能会与客户所量测的大小会很近似。如果量测的长度与实际长度有冲突，那么"GAPDETECT"指令将不会起作用。可以通过设置指令中的参数来侦测有预印图案及文字的标签。

函数构造：

```
public static byte[] gapDetect(int x, int y)
```

```
public static byte[] gapDetect()
```

参数：

x - 纸张高度 (单位 dots)

y - 间隙高度 (单位 dots)

若无 x, y 参数, 打印机将自行检测。

返回:

byte[]

blineDetect

说明:

此条指令会使打印机持续出纸, 通过纸张及黑标通过黑标感测器时所侦测到的强度差异, 来区分定义纸张及黑标的大小。一般状况下使用此条指令所侦测到的纸张大小及黑标大小可能会与客户所量测的大小会很近似。如果量测的长度与实际长度有冲突, 那么 "BLINEDetect" 指令将不会起作用。可以通过设置指令中的参数来侦测有预印图案及文字的标签。

函数构造:

```
public static byte[] blineDetect(int x, int y)
public static byte[] blineDetect()
```

参数:

x - 纸张高度 (单位 dots)

y - 间隙高度 (单位 dots)

若无 x, y 参数, 打印机将自行检测。

返回:

byte[]

autoDetect

说明:

此条指令会使打印机持续出纸, 通过纸张及间隙/黑标通过间隙感测器/黑标感测器时所侦测到的强度差异, 来区分定义纸张及黑标的大小。一般状况下使用此条指令所侦测到的纸张大小及间隙/黑标大小可能会与客户所量测的大小会很近似。如果量测的长度与实际长度有冲突, 那么 "AUTODETECT" 指令将不会起作用。可以通过设置指令中的参数来侦测有预印图案及文字的标签。

函数构造:

```
public static byte[] autoDetect(int x, int y)
public static byte[] autoDetect()
```

参数:

x - 纸张高度 (单位 dots)

y - 间隙高度 (单位 dots)

若无 x, y 参数, 打印机将自行检测。

如果使用此条指令, 那么打印机会使用适当的感测器来侦测纸张, 请勿在程式中再使用 "GAP" 或者 "BLINE" 指令。

此条指令仅在 V6.86 及以后版本 Firmware 中支持。

返回:

byte[]

blineByinch

说明：

定义两张纸中间的黑标高度及偏移距离，单位英寸（inch）

函数构造：

```
public static byte[] blineByinch(double m, double n)
```

参数：

m - 黑标高度

$0 \leq m \leq 1$ (inch), $0 \leq m \leq 25.4$ (mm) $0 \leq m \leq 5$ (inch),

$0 \leq m \leq 127$ (mm) /仅在V6.27及以后版本Firmware

n - 黑标偏移距离

$0 \leq n \leq \text{label length}$

返回：

byte[]

blineBymm

说明：

定义两张纸中间的黑标高度及偏移距离，单位毫米（mm）

函数构造：

```
public static byte[] blineBymm(double m, double n)
```

参数：

m - 黑标高度

$0 \leq m \leq 1$ (inch), $0 \leq m \leq 25.4$ (mm) $0 \leq m \leq 5$ (inch),

$0 \leq m \leq 127$ (mm) /仅在V6.27及以后版本Firmware

n - 黑标偏移距离

$0 \leq n \leq \text{label length}$

返回：

byte[]

blineBydot

说明：

定义两张纸中间的黑标高度及偏移距离，单位像素（dot）。

仅在 V6.27 及以后版本 Firmware 中支持。

函数构造：

```
public static byte[] blineBydot(int m, int n)
```

参数：

m - 黑标高度

$0 \leq m \leq 1$ (inch), $0 \leq m \leq 25.4$ (mm) $0 \leq m \leq 5$ (inch),

$0 \leq m \leq 127$ (mm) /仅在V6.27及以后版本Firmware

n - 黑标偏移距离

$0 \leq n \leq \text{label length}$

返回:

byte[]

offsetByinch

说明:

此条指令将为每一张标签定义额外的送纸偏移,对于剥纸模式及裁切模式下,调整此条指令,可以使打印机将纸张停留在预期的位置上。打印机在每次打印前会将额外设定的距离回卷后再打印。单位英寸 (inch)

不正确的偏移参数有可能会造成纸张卡纸

函数构造:

public static byte[] offsetByinch(double m)

参数:

m - 偏移距离

$0 \leq m \leq 1$ (inch),

返回:

byte[]

offsetBymm

说明:

此条指令将为每一张标签定义额外的送纸偏移,对于剥纸模式及裁切模式下,调整此条指令,可以使打印机将纸张停留在预期的位置上。打印机在每次打印前会将额外设定的距离回卷后再打印。单位毫米 (mm)

不正确的偏移参数有可能会造成纸张卡纸

函数构造:

public static byte[] offsetBymm(double m)

参数:

m - 偏移距离

$0 \leq m \leq 1$ (inch)

返回:

byte[]

offsetBydot

说明：

此条指令将为每一张标签定义额外的送纸偏移，对于剥纸模式及裁切模式下，调整此条指令，可以使打印机将纸张停留在预期的位置上。打印机在每次打印前会将额外设定的距离回卷后再打印。单位毫米（mm）

不正确的偏移参数有可能会造成纸张卡纸。

仅在 V6.27 及以后版本 Firmware 中支持。

函数构造：

```
public static byte[] offsetBydot(int m)
```

参数：

m - 黑标高度

$0 \leq m \leq 1$ (inch), $0 \leq m \leq 25.4$ (mm) $0 \leq m \leq 5$ (inch),

$0 \leq m \leq 127$ (mm) /仅在V6.27及以后版本Firmware

n - 黑标偏移距离

$0 \leq n \leq \text{label length}$

返回：

byte[]

speed

说明：

用来定义打印机的打印速度。

函数构造：

```
public static byte[] speed(double n)
```

参数：

n - 打印速度为 n inch/s

返回：

byte[]

density

说明：

用来定义打印时的浓度。

默认的DENSITY设置为 8.

函数构造：

```
public static byte[] density(int n)
```

参数：

n - 0~15

0: 制定最浅的浓度

15: 指定为最深的浓度

返回：

byte[]

direction

说明:

此条指令将会定义打印机的打印方向及内容是否镜面反射,此设置将会保存在打印机的内存中。

函数构造:

```
public static byte[] direction(int n)
public static byte[] direction(int n, int m)
```

参数:

n - 0 or 1.
m - 0: 正常的图片 1: 镜面反射后的图片

返回:

byte[]

reference

说明:

此条指令用来定义标签打印的起始参考点。参考原点的位置将会收到打印方向的影响。

函数构造:

```
public static byte[] reference(int x, int y)
```

参数:

x -横坐标 (单位 DOT)
y -纵坐标 (单位 DOT)

返回:

byte[]

shift

说明:

此条指令用来调整标签图像的横坐标及纵坐标。数值为正值时, 会向标签打印的方向移动; 负值时, 会背离所打印的方向。

函数构造:

```
public static byte[] shift(int x)
public static byte[] shift(int x, int y)
```

参数:

x -可选.最大可设置尺寸为 1 inch。对于 203DPI 的机器, 范围是 -203 到 203, 对于 300DPI,范围是-300 到 300, 单位为 DOT。

y -最大可设置尺寸为 1 inch。对于 203DPI的机器,范围是 -203 到 203,对于 300DPI,范围是-300 到 300, 单位为 DOT。

返回：

byte[]

country

说明：

此条指令用来定义接收键盘上输入的不同国家的特殊字符。

函数构造：

```
public static byte[] country(java.lang.String n)
```

参数：

n - 001: USA
002: Canadian-French
003: Spanish (Latin America)
031: Dutch
032: Belgian
033: French (France)
034: Spanish (Spain)
036: Hungarian
038: Yugoslavian
039: Italian
041: Switzerland
042: Slovak
044: United Kingdom
045: Danish
046: Swedish
047: Norwegian
048: Polish
049: German
055: Brazil
061: English (International)
351: Portuguese
358: Finnish

返回：

byte[]

codePage

说明：

此条指令用来定义不同的国码编码。

函数构造：

```
public static byte[] codePage(java.lang.String n)
```

参数:

n - 对应 CODEPAGE 的名称或代码。

返回:

byte[]

cls**说明:**

此条指令用来清除打印机中的图像缓存。

此条指令必须在 SIZE 后进行设定。

函数构造:

```
public static byte[] cls()
```

参数:

无

返回:

byte[]

feed**说明:**

此条指令用来命令打印机送出规定长度的纸张，长度的数值单位为 DOT

函数构造:

```
public static byte[] feed(int n)
```

参数:

n - 单位: dot

$1 \leq n \leq 9999$

返回:

byte[]

backFeed**说明:**

此条指令用来命令打印机回退规定长度的纸张，长度的数值单位为 DOT。

警告：不正确的回纸距离有可能会造成纸张卡纸或者褶皱。

函数构造:

```
public static byte[] backFeed(int n)
```

参数:

n - 单位: dot

$1 \leq n \leq 9999$

返回:

byte[]

formFeed

说明:

用来将纸张推送到下张标签的起始位置。
此条指令必须被放置在 **SIZE** 指令之后。

函数构造:

```
public static byte[] formFeed()
```

参数:

无

返回:

byte[]

home

说明:

此条指令会使打印机持续出纸，直到打印机感测器侦测到纸张的起始位置。
此条指令必须要在 **SIZE** 及 **GAP** 之后使用
支持 **TSPL** 的打印机: 退回标签到起始位置
支持 **TSPL2** 的打印机: 送出标签到起始位置

函数构造:

```
public static byte[] home()
```

参数:

无

返回:

byte[]

print

说明:

用于将打印机中图像缓存印出

函数构造:

```
public static byte[] print(int m, int n)
```

```
public static byte[] print(int m)
```

参数:

m -如设置了序列号，指定列印多少序列号的标签 $1 \leq m \leq 999999999$ ，否则指定打印多少份

n -指定每一张序列号打印多少份 $1 \leq n \leq 999999999$

返回:

byte[]

sound

说明：

此条指令用来控制蜂鸣器所发出的声音频率，声音大小分十级，可用”Interval”参数来控制频率。

函数构造：

```
public static byte[] sound(int level, int interval)
```

参数：

level -声音大小: 0~9

interval -声音间隔: 1~4095

返回：

byte[]

cut

说明：

此条指令会使打印机立刻裁切一次，并不执行回拉动作。

函数构造：

```
public static byte[] cut()
```

参数：

无

返回：

byte[]

limitFeedBymm

说明：

限定定间隙校正执行的最大长度，若在此长度范围内无法测得间隙存在，则将传感器模式定为连续纸模式。单位为毫米（mm）。

函数构造：

```
public static byte[] limitFeedBymm(double n)
```

参数：

n -最小的纸张侦测长度

返回：

byte[]

limitFeedByinch

说明：

限定定间隙校正执行的最大长度，若在此长度范围内无法测得间隙存在，则将传感器模式定为连续纸模式。单位为英寸（inch）。

函数构造：

```
public static byte[] limitFeedByinch(double n)
```

参数:

n -最小的纸张侦测长度

返回:

byte[]

limitFeedBydot

说明:

限定定间隙校正执行的最大长度，若在此长度范围内无法测得间隙存在，则将传感器模式定为连续纸模式。单位为像素（dot）。

函数构造:

```
public static byte[] limitFeedBydot(int n)
```

参数:

n -最小的纸张侦测长度

返回:

byte[]

selfTest

说明:

此条指令将立即打印自检信息。

函数构造:

```
public static byte[] selfTest()
```

```
public static byte[] selfTest(java.lang.String page)
```

参数:

page - 缺省: 打印整个自检信息。

PATTERN: 打印一块灰色图案来测试打印头状态。

ETHERNET: 打印一张打印机的网络设定信息。

WLAN: 打印一张打印机的 WIFI 设定信息。

RS232: 打印一张打印机的 RS232 端口信息。

SYSTEM: 打印一张打印机的当前设定信息。

Z: 打印一张当前打印机的仿真语言设定信息。

返回:

byte[]

ej

说明:

打印机会将位于此条指令之前的命令执行完毕后，再去执行后方的指令。

注意: 此条指令仅在 V6.39 及以后版本 Firmware 中支持。

函数构造:

```
public static byte[] eoj()
```

参数:

无

返回:

byte[]

delay

说明:

此条指令会让打印机等待指定的时间，然后才去执行后方的指令。

函数构造:

```
public static byte[] delay(int ms)
```

参数:

ms - 指定时间长短。单位 millisecond. 1000 ms = 1 second.

返回:

byte[]

disPlay

说明:

用来定义打印时的浓度。

函数构造:

```
public static byte[] disPlay(java.lang.String s)
```

参数:

s - IMAGE: 将打印机缓存中的图像显示在显示屏上。 (V6.39 EZ 之后)

返回:

byte[]

initialPrinter

说明:

此条指令会将打印机恢复至出厂设置。

函数构造:

```
public static byte[] initialPrinter()
```

参数:

无

返回:

byte[]

bar

说明：

汇制线条。

函数构造：

```
public static byte[] bar(int x, int y, int width, int height)
```

参数：

x -左上角的 X 坐标 (单位 dot)

y -左上角的 Y 坐标 (单位 dot)

width -线条宽度 (单位 dot)

height -线条高度 (单位 dot)

返回：

byte[]

barCode

说明：

在标签中绘制一维条码。可用的条码类型查看编程手册

函数构造：

```
public static byte[] barCode(int x, int y, java.lang.String codeType, int height, int human,
int rotation, int narrow, int wide, java.lang.String content)
```

参数：

x - 指定条码 x 坐标

y -指定条码 y 坐标

codeType - 条码类型

height - 条码高度（单位 dot）

human - 0: 不显示码文

1: 显示码文，码文左对齐

2: 显示码文，码文居中

3: 显示码文，码文右对齐

rotation - 0：不旋转

90：顺时针旋转 90 度

180：顺时针旋转 90 度

270：顺时针旋转 270 度

narrow -窄比例因子宽度 (单位 dot)

wide -宽比例因子宽度 (单位 dot)

content - 条码内容

返回：

byte[]

bitmap

说明：

绘制 bitmap 图像。

函数构造:

```
public static byte[] bitmap(int x, int y, int mode, Bitmap bitmap,
BitmapToByteData.BmpType bmpType , int paperWidth, int paperHeight)
```

参数:

x - 指定 x 坐标

y - 指定 y 坐标

width - 图片宽度 (单位 byte)

height - 图片高度 (单位 dots)

mode - 图像格式:

0: OVERWRITE

1: OR

2: XOR

bitmap - Bitmap 数据

bmpType - 图片处理类型

paperWidth - 标签宽度 (单位 dots)

paperHeight - 标签高度 (单位 dots), 高度为图片打印的最大长度, 设置错误可能导致打印失败。

返回:

byte[]

box

说明:

绘制方框。

函数构造:

```
public static byte[] box(int x, int y, int x_end, int y_end, int thickness)
```

```
public static byte[] box(int x, int y, int x_end, int y_end, int thickness, int radius)
```

参数:

x -指定左上角的 X 坐标 (单位 DOT)

y -指定左上角的 Y 坐标 (单位 DOT)

x_end -指定右上角的 X 坐标 (单位 DOT)

y_end -指定右上角的 Y 坐标 (单位 DOT)

thickness -线条粗细 (单位 DOT)

radius - 可选, 指定圆角半径单位 (DOT), 默认为 0

返回:

byte[]

circle

说明:

绘制一个圆形。

函数构造:

```
public static byte[] circle(int x_start, int y_start, int diameter, int thickness)
```

参数:

x_start -指定左上角的 X 坐标 (单位 DOT)

y_start -指定左上角的 Y 坐标 (单位 DOT)

diameter -指定圆的直径 (单位 DOT)

thickness -线条粗细 (单位 DOT)

返回:

byte[]

ellipse

说明:

绘制一个椭圆。

函数构造:

```
public static byte[] ellipse(int x, int y, int width, int height, int thickness)
```

参数:

x -指定左上角的 X 坐标 (单位 DOT)

y -指定左上角的 Y 坐标 (单位 DOT)

width -指定椭圆的宽 (单位 DOT)

height -指定椭圆的高 (单位 DOT)

thickness -线条粗细 (单位 DOT)

返回:

byte[]

codaBlockFMode

说明:

绘制 CODABLOCK F 类型条码。

函数构造:

```
public static byte[] codaBlockFMode(int x, int y, int rotation, int row_height, int module_width, String content)
```

参数:

x -指定 X 坐标 (单位 DOT)

y -指定 Y 坐标 (单位 DOT)

rotation - 0 : 不旋转

90 : 顺时针旋转 90 度

180 : 顺时针旋转 90 度

270 : 顺时针旋转 270 度

row height - 模块的高度 (默认为 8)

module width - CODABLOCK 窄条码比例因子(单位 DOT, 默认为 2)

content - CODABLOCK 条码内容 (单位 DOT)

返回:
byte[]

dmatrix

说明:
绘制 DataMatrix 二维条码，当前仅支持 ECC200 纠正种类。
此条指令仅在 V6.89EZ 及以后版本 Firmware 中支持。

函数构造:

```
public static byte[] dmatrix(int x, int y, int width, int height, int xm, int row, int col,
String content)
```

```
public static byte[] dmatrix(int x, int y, int width, int height, int xm, String content)
```

参数:

x -指定 X 坐标 (单位 DOT)
y -指定 Y 坐标 (单位 DOT)
xm - 模块大小(单位 DOT)
row - 行码: 10 to 144
col - 栏码: 10 to 144
content -DataMatrix 数据内容

返回:
byte[]

erase

说明:
清除对应区域图像缓存。

函数构造:

```
public static byte[] erase(int x, int y, int width, int height)
```

参数:

x -指定 X 起始坐标 (单位 DOT)
y -指定 Y 起始坐标 (单位 DOT)
width - x 方向的区域宽度(单位 DOT)
height - y 方向的区域宽度(单位 DOT)

返回:
byte[]

pdf417

说明:
绘制 PDF417 二维条码。

函数构造:

```
public static byte[] pdf417(int x, int y, int width, int height, int rotate, String option,
String content)
```

参数:

x - 指定 pdf417 的 x 坐标

y -指定 pdf417 的 y 坐标

width - 条码期望宽度 (单位 dot)

height - 条码期望高度 (单位 dot)

rotation - 0 : 不旋转

90 : 顺时针旋转 90 度

180 : 顺时针旋转 90 度

270 : 顺时针旋转 270 度

option - 参数请查看编程手册, 设置多个以, 隔开, 例如"E4,W4,T1";

content - 条码内容

返回:

byte[]

putBmp

说明:

打印 BMP 格式的图片。只有热敏型打印机支持灰度打印, 且仅支持 1-bit (单色图) 和 8-bit (256 色) BMP 图片。

函数构造:

```
public static byte[] putBmp(int x, int y, String filename, int bpp, int contrast)
```

```
public static byte[] putBmp(int x, int y, String filename)
```

参数:

x -指定 X 坐标 (单位 DOT)

y -指定 Y 坐标 (单位 DOT)

filename -所下载的 BMP 图片名称 (支持 ZPL *.GRF 档案)

bpp -选填, 灰度打印, 默认为 1 *V6.91EZ 之后

1: 1-bit (单色图)

8: 8-bit (256 色)图片

contrast -选填, 灰度对比率 默认为 80 建议选择范围在 60 至 100 之间。

*V6.91EZ 之后。

返回:

byte[]

putpcx

说明:

打印 PCX 格式图片。TSPL 语言可支持 2 色的 PCX 图片, TSPL2 语言可支持 256

色 PCX 图片。

函数构造：

```
public static byte[] putpcx(int x, int y, String filename)
```

参数：

x -指定 X 坐标 (单位 DOT)

y -指定 Y 坐标 (单位 DOT)

filename -所下载的 PCX 图片的名称（区分大小写）

返回：

byte[]

qrCode

说明：

绘制 QR code。

函数构造：

```
public static byte[] qrCode(int x, int y, String eccLevel, int cellWidth, String mode, int rotation, String model, String mask, String content)
```

参数：

x - 指定 qrcode 的 x 坐标

y -指定 qrcode 的 y 坐标

eccLevel - 纠错等级

L: 不旋转

M：顺时针旋转 90 度

Q：顺时针旋转 90 度

270：顺时针旋转 270 度

cellWidth - 模块大小 1~10

mode - 自动/手动编码

A: 自动

M: 手动

rotation - 0：不旋转

90：顺时针旋转 90 度

180：顺时针旋转 90 度

270：顺时针旋转 270 度

model - M1: 默认，原始版本

M2: 扩大版本（大部分智能手机支持此版本）

mask - S0~S8，默认为 S7

content - 条码内容

返回：

byte[]

reverse

说明:

将指定区域的图像缓存反白。

函数构造:

```
public static byte[] reverse(int x, int y, int width, int height)
```

参数:

x -指定 X 坐标 (单位 DOT)

y -指定 Y 坐标 (单位 DOT)

width - x 方向上的宽度 (单位 DOT)

height - y 方向上的高度 (单位 DOT)

返回:

byte[]

text

说明:

打印一段文字。

函数构造:

```
public static byte[] text(int x, int y, String font, int rotation, int x_multiplication, int y_multiplication, String content)
```

参数:

x -指定 X 坐标 (单位 DOT)

y -指定 Y 坐标 (单位 DOT)

font - 字体名称 (单位 DOT)

rotation - 0 : 不旋转

90 : 顺时针旋转 90 度

180 : 顺时针旋转 90 度

270 : 顺时针旋转 270 度

x_multiplication -水平放大比率, 最大放大 10 倍 有效参数: 1~10 对于矢量字体设置, 此参数无此限制

y_multiplication -垂直放大比率,最大放大 10 倍

content - 文本内容

返回:

byte[]

block

说明:

用来定义打印时的浓度。

函数构造:

```
public static byte[] block(int x, int y, int width, int height, String font, int rotation, int x_multiplication, int y_multiplication, int space, int alignment, String content)
```

```
public static byte[] block(int x, int y, int width, int height, String font, int rotation, int
```

x_multiplication, int y_multiplication, String content)

参数:

x -指定 X 坐标 (单位 DOT)

y -指定 Y 坐标 (单位 DOT)

width - 段落的宽度 (单位 DOT)

height - 段落的高度 (单位 DOT)

font - 字体名称 (单位 DOT)

rotation - 0 : 不旋转

90 : 顺时针旋转 90 度

180 : 顺时针旋转 90 度

270 : 顺时针旋转 270 度

x_multiplication -水平放大比率, 最大放大 10 倍 有效参数: 1~10 对于矢量字体设置, 此参数无此限制

y_multiplication -垂直放大比率,最大放大 10 倍

space -在每一行中间添加或者删除空格 单位 DOT。

alignment - 文本位置

0: 默认 (居左)

1: 居左

2: 居中

3: 居右

content - 文本内容

返回:

byte[]

checkPrinterStateByPort9100

说明:

此指令使用于 9100 端口, 发一次, 每当打印机状态改变, 就有数据返回。

函数构造:

```
public static byte[] checkPrinterStateByPort9100()
```

参数:

无

返回:

byte[]

checkPrinterStateByPort4000

说明:

此指令使用于 4000 端口, 发一次, 打印机返回一次代表状态的数据。

函数构造:

```
public static byte[] checkPrinterStateByPort4000()
```

参数:

无 返回： byte[]

9. DataForSendToPrinterPos58

说明：

该类处理的指令只适用于 80mm 系列热敏打印机 XP—58xx 系列的产品； 具体指令的描述和效果请参考编程手册。具体功能参考 DataForSendToPrinterTSC。

目录：

net.posprinter.utils.DataForSendToPrinterPos58

10. DataForSendToPrinterPos76

说明：

本列为指令处理工具类。该类处理的指令只适用于 76mm 系列针打打印机 pos—76xx 系列的产品； 具体指令的描述和效果请参考编程手册。具体功能参考 DataForSendToPrinterTSC。

目录：

net.posprinter.utils.DataForSendToPrinterPos76

11. DataForSendToPrinterPos80

说明：

本列为指令处理工具类。该类处理的指令只适用于 80mm 系列热敏打印机 pos—80xx 系列的产品； 具体指令的描述和效果请参考编程手册。具体功能参考 DataForSendToPrinterTSC。

目录：

net.posprinter.utils.DataForSendToPrinterPos80

12. PosPrinterDev

说明：

打印机收发信息处理

目录:

net.posprinter.utils.PosPrinterDev

● 枚举类型:

portType

说明:

接口类型。

变量:

Unknown: 未知;

USB: USB;

Bluetooth: 蓝牙;

Ethernet: 网口。

ErrorCode

说明:

返回数据。

变量:

OpenPortFailed,
OpenPortSuccess,
ClosePortFailed,
ClosePortSuccess,
WriteDataFailed,
WriteDataSuccess,
ReadDataSuccess,
ReadDataFailed,
UnknownError

● 构造器详细资料:

```
public PosPrinterDev(PortType portType, Context context)
public PosPrinterDev(PortType portType, Context context, String usbPathName)
public PosPrinterDev(PortType portType, String bluetoothID)
public PosPrinterDev(PortType portType, String ethernetIP, int ethernetPort)
```

● 方法详细资料:

open

说明：

开启收发功能。

函数构造：

```
public ReturnMessage Open()
```

参数：

无

返回：

ReturnMessage

write

说明：

发送数据。

函数构造：

```
public ReturnMessage Write(int data)
```

```
public ReturnMessage Write(byte[] data)
```

```
public ReturnMessage Write(byte[] data, int offset, int count)
```

参数：

data - 数据；

offset - data 的起始位置；

count - data 计数。

返回：

ReturnMessage

read

说明：

读取接收到的数据。

函数构造：

```
public ReturnMessage Read(byte[]buffer)
```

```
public ReturnMessage Read(byte[]buffer, int offset, int count)
```

```
public int Read()
```

参数：

buffer - 数据流；

offset - 数据的起始位置；

count - 数据计数。

返回：

ReturnMessage

close

说明:

关闭收发功能。

函数构造:

```
public synchronized ReturnMessage Close()
```

参数:

无

返回:

ReturnMessage

GetPortInfo

说明:

查看当前端口信息。

函数构造:

```
public PortInfo GetPortInfo()
```

参数:

无

返回:

端口信息

GetUsbPathNames

说明:

查看当前连接的 USB 设备名称。

函数构造:

```
public static List<String> GetUsbPathNames(Context context)
```

参数:

context - 环境对象

返回:

USB 设备名称

13. ReadExcel

说明:

读取 excel 文件中的数据

目录:

net.posprinter.utils.ReadExcel

●方法详细资料:

readExcel

说明:

读取 excel 文件中的数据。

函数构造:

```
public static HashMap<String, String[]> readExcel(Context context, String path, int sheetNum)
```

参数:

context - 环境对象

path - 文件路径

sheetNum - 工作表的编号

返回:

HashMap<String, String[]> - 指定工作表中的全部数据。

String: 表头;

String[]: 对应表头中的全部内容。

getExcelSheetName

说明:

读取 excel 文件中的数据。

函数构造:

```
public static String[] getExcelSheetName(Context context, String path)
```

参数:

context - 环境对象

path - 文件路径

返回:

String[] 所有的工作表

14. RoundQueue

说明:

用来存储冲打印机发送过来的数据而设计的数据类型

目录:

net.posprinter.utils.RoundQueue

●构造器详细资料:

```
public RoundQueue(int size)
```

●方法详细资料：

addLast

说明：

向尾部添加一个元素。

函数构造：

```
public void addLast(T element)
```

参数：

element - 数据 byte[]

返回：

无

removeFirst

说明：

移除第一个元素。

函数构造：

```
public T removeFirst()
```

参数：

无

返回：

被移除的数据

realSize

说明：

查询队列中的数据数量。

函数构造：

```
public int realSize()
```

参数：

无

返回：

int - 队列中的数据数量

isEmpty

说明：

判断队列是否为空。

函数构造：

```
public void isEmpty()
```

参数：

无

返回：

boolean

isFull

说明：

判断队列是否已满。

函数构造：

```
public void isFull()
```

参数：

无

返回：

boolean

clear

说明：

清除保存的所有数据。

函数构造：

```
public void clear()
```

参数：

无

返回：

无

get

说明：

返回指定位置的数据

函数构造：

```
public T get(int index)
```

参数：

index - 位置（int）

返回：

指定数据

indexOf

说明:

获取元素在队列中的索引。

函数构造:

```
public int indexOf(T key)
```

参数:

key - 指定元素数据

返回:

找到就返回其位置，找不到就返回-1

如果 key 为 null 则永远返回-1

gethead

说明:

得到环形队列中头游标的位置，代表队列中最先进入的数据的索引

函数构造:

```
public int gethead()
```

参数:

无

返回:

int

gettail

说明:

得到环形队列中尾游标的位置，代表最后进入队列的索引

函数构造:

```
public int gettail()
```

参数:

无

返回:

int

getLast

说明:

得到环形队列尾部元素，也就是最后进入队列的数据

函数构造:

```
public T getLast()
```

参数:

无
返回：
指定数据

getFirst

说明：
获得环形队列最先进入队列的元素
函数构造：
`public T getFirst()`
参数：
无
返回：
指定数据