

## 历史记录

版本	修改描述	时间
1.0.0	初始版本，重新编写文档	2019.04.03

### 1.枚举

- 1.1 纸张类型
- 1.2 标签分辨率
- 1.3 条码类型
- 1.4 条码宽窄条比例
- 1.5 二维码规范编号
- 1.6 二维码单元宽度
- 1.7 二维码纠错水平
- 1.8 二维码字符类型
- 1.9 文本的字体类型
- 1.10 缩放文本字体名称
- 1.11 字体组编号
- 1.12 字体加粗
- 1.13 旋转角度
- 1.14 字体放大系数
- 1.15 泰文字体类型
- 1.16 PDF417码的纠错等级
- 1.17 PDF417码的方向
- 1.18 PDF417数据压缩模式

### 2.属性

- 2.1 cmdData
- 2.2 encoding

### 3.接口

- 3.1 添加指令文本
- 3.2 添加指令数据
- 3.3 打印自检页
- 3.4 获取打印机状态
- 3.5 打印完成状态的开关
- 3.6 打开下划线
- 3.7 关闭下划线
- 3.8 获取SN号
- 3.9 设置纸张类型
- 3.10 初始化标签
- 3.11 指定标签宽度
- 3.12 一维条码
  - 3.12.1 横向条码
  - 3.12.2 纵向条码

- 3.12.3 条码下方文字
- 3.12.4 结束条码底部文字展示
- 3.13 二维码
  - 3.13.1 横向QR码
  - 3.13.2 纵向QR码
  - 3.13.3 添加QR码数据
  - 3.13.4 结束QR码
- 3.14 PDF417码
- 3.15 绘制矩形
- 3.16 绘制线条
- 3.17 绘制反白线条
- 3.18 打印图片
- 3.19 串联文本
- 3.20 文本
  - 3.20.1 打印文本
  - 3.20.2 反白加粗文本
  - 3.20.3 居中文本
  - 3.20.4 自动换行文本
  - 3.20.5 自动换行居中文本
  - 3.20.6 泰文自动换行
- 3.21 对齐方式
  - 3.21.1 左对齐
  - 3.21.2 居中
  - 3.21.3 右对齐
- 3.22 打印标签
- 3.23 反向打印标签
- 3.24 多行文本
- 3.25 缩放字体文本
- 3.26 字体加粗
- 3.27 字间距
- 3.28 字体放大倍数
- 3.29 水印文本
- 3.30 左边距
- 3.31 走纸
- 3.32 打印速度
- 3.33 定位缝标
- 3.34 打印浓度
- 3.35 行模式开启
- 3.36 行模式指令前缀
- 3.37 关闭倒带
- 3.38 设置字符代码页
- 3.39 阿拉伯语变形
- 3.40 泰文变形
- 3.41 越南语变形
- 3.42 获取固件版本

# 1.枚举

## 1.1 纸张类型

- 代码

```
typedef NS_ENUM(NSUInteger, PTCPCLPaperType) {
    PTCPCLPaperContinuoue      = 0,    ///< continuous paper
    PTCPCLPaperLabel          = 1,    ///< label paper
    PTCPCLPaperBackBlackMark  = 2,    ///< back black mark
    PTCPCLPaperFrontBlackMark = 3,    ///< front black mark
    PTCPCLPaperBlackMarkInch3 = 4,    ///< 3 inch black mark
    PTCPCLPaperBlackMarkInch2 = 5,    ///< 2 inch black mark
};
```

- 说明

枚举成员	枚举值	描述
PTCPCLPaperContinuoue	0	连续纸
PTCPCLPaperLabel	1	标签纸
PTCPCLPaperBackBlackMark	2	后黑标纸
PTCPCLPaperFrontBlackMark	3	前黑标纸
PTCPCLPaperBlackMarkInch3	4	3寸黑标纸
PTCPCLPaperBlackMarkInch2	5	2寸黑标纸

## 1.2 标签分辨率

- 代码

```
typedef NS_ENUM(NSUInteger, PTCPCLLabelResolution) {
    PTCPCLLabelResolution100 = 100,    ///< 100
    PTCPCLLabelResolution200 = 200,    ///< 200
};
```

- 说明

枚举成员	枚举值	描述
PTCPCLLabelResolution100	100	100dpi的标签分辨率
PTCPCLLabelResolution200	200	200dpi的标签分辨率

## 1.3 条码类型

- 代码

```
typedef NS_ENUM(NSUInteger, PTCPClBarcodeStyle) {
    PTCPClBarcodeStyleUPCA      = 0,
    PTCPClBarcodeStyleUPCE     = 1,
    PTCPClBarcodeStyleEAN13    = 2,
    PTCPClBarcodeStyleEAN8     = 3,
    PTCPClBarcodeStyleCode39   = 4,
    PTCPClBarcodeStyleCode93   = 5,
    PTCPClBarcodeStyleCode128  = 6,
    PTCPClBarcodeStyleCodeBAR  = 7
};
```

- 说明

枚举成员	枚举值	描述
PTCPCLBarcodeStyleUPCA	0	UPCA类型的条码
PTCPCLBarcodeStyleUPCE	1	UPCE类型的条码
PTCPCLBarcodeStyleEAN13	2	EAN13类型的条码
PTCPCLBarcodeStyleEAN8	3	EAN8类型的条码
PTCPCLBarcodeStyleCode39	4	Code39类型的条码
PTCPCLBarcodeStyleCode93	5	Code93类型的条码
PTCPCLBarcodeStyleCode128	6	Code128类型的条码
PTCPCLBarcodeStyleCodeBAR	7	BAR类型的条码

## 1.4 条码宽窄条比例

- 代码

```
typedef NS_ENUM(NSUInteger, PTCPCLBarcodeBarRatio) {  
    PTCPCLBarcodeBarRatio0      = 0,  
    PTCPCLBarcodeBarRatio1      = 1,  
    PTCPCLBarcodeBarRatio2      = 2,  
    PTCPCLBarcodeBarRatio3      = 3,  
    PTCPCLBarcodeBarRatio4      = 4,  
    PTCPCLBarcodeBarRatio20     = 20,  
    PTCPCLBarcodeBarRatio21     = 21,  
    PTCPCLBarcodeBarRatio22     = 22,  
    PTCPCLBarcodeBarRatio23     = 23,  
    PTCPCLBarcodeBarRatio24     = 24,  
    PTCPCLBarcodeBarRatio25     = 25,  
    PTCPCLBarcodeBarRatio26     = 26,  
    PTCPCLBarcodeBarRatio27     = 27,  
    PTCPCLBarcodeBarRatio28     = 28,  
    PTCPCLBarcodeBarRatio29     = 29,  
    PTCPCLBarcodeBarRatio30     = 30  
};
```

- 说明

枚举成员	枚举值	描述
PTCPCLBarcodeBarRatio0	0	1.5:1
PTCPCLBarcodeBarRatio1	1	2.0:1
PTCPCLBarcodeBarRatio2	2	2.5:1
PTCPCLBarcodeBarRatio3	3	3.0:1
PTCPCLBarcodeBarRatio4	4	3.5:1
PTCPCLBarcodeBarRatio20	20	2.0:1
PTCPCLBarcodeBarRatio21	21	2.1:1
PTCPCLBarcodeBarRatio22	22	2.2:1
PTCPCLBarcodeBarRatio23	23	2.3:1
PTCPCLBarcodeBarRatio24	24	2.4:1
PTCPCLBarcodeBarRatio25	25	2.5:1
PTCPCLBarcodeBarRatio26	26	2.6:1
PTCPCLBarcodeBarRatio27	27	2.7:1
PTCPCLBarcodeBarRatio28	28	2.8:1
PTCPCLBarcodeBarRatio29	29	2.9:1
PTCPCLBarcodeBarRatio30	30	3.0:1

## 1.5 二维码规范编号

- 代码

```
typedef NS_ENUM(NSUInteger, PTCPLQRCodeMode1) {
    PTCPLQRCodeMode11    = 1,
    PTCPLQRCodeMode12    = 2    ///< default
};
```

- 说明

枚举成员	枚举值	描述
PTCPCLQRCodeModel1	1	原始规范
PTCPCLQRCodeModel2	2	推荐规范，提供了附加功能，默认值

## 1.6 二维码单元宽度

- 代码

```
typedef NS_ENUM(NSUInteger, PTCPLQRCodeUnitwidth) {
    PTCPLQRCodeUnitwidth_1 = 1,
    PTCPLQRCodeUnitwidth_2,
    PTCPLQRCodeUnitwidth_3,
    PTCPLQRCodeUnitwidth_4,
    PTCPLQRCodeUnitwidth_5,
    PTCPLQRCodeUnitwidth_6,           ///< default
    PTCPLQRCodeUnitwidth_7,
    PTCPLQRCodeUnitwidth_8,
    PTCPLQRCodeUnitwidth_9,
    PTCPLQRCodeUnitwidth_10,
    PTCPLQRCodeUnitwidth_11,
    PTCPLQRCodeUnitwidth_12,
    PTCPLQRCodeUnitwidth_13,
    PTCPLQRCodeUnitwidth_14,
    PTCPLQRCodeUnitwidth_15,
    PTCPLQRCodeUnitwidth_16,
    PTCPLQRCodeUnitwidth_17,
    PTCPLQRCodeUnitwidth_18,
    PTCPLQRCodeUnitwidth_19,
    PTCPLQRCodeUnitwidth_20,
    PTCPLQRCodeUnitwidth_21,
    PTCPLQRCodeUnitwidth_22,
    PTCPLQRCodeUnitwidth_23,
    PTCPLQRCodeUnitwidth_24,
    PTCPLQRCodeUnitwidth_25,
    PTCPLQRCodeUnitwidth_26,
    PTCPLQRCodeUnitwidth_27,
    PTCPLQRCodeUnitwidth_28,
    PTCPLQRCodeUnitwidth_29,
    PTCPLQRCodeUnitwidth_30,
    PTCPLQRCodeUnitwidth_31,
    PTCPLQRCodeUnitwidth_32
};
```

- 说明

枚举成员	枚举值	描述
PTCPCLQRCodeUnitWidth_1	1	1个像素点
PTCPCLQRCodeUnitWidth_2	2	2
PTCPCLQRCodeUnitWidth_3	3	3
PTCPCLQRCodeUnitWidth_4	4	4
PTCPCLQRCodeUnitWidth_5	5	5
PTCPCLQRCodeUnitWidth_6	6	默认值
PTCPCLQRCodeUnitWidth_7	7	7
PTCPCLQRCodeUnitWidth_8	8	8
PTCPCLQRCodeUnitWidth_9	9	9
PTCPCLQRCodeUnitWidth_10	10	10
PTCPCLQRCodeUnitWidth_11	11	11
PTCPCLQRCodeUnitWidth_12	12	12
PTCPCLQRCodeUnitWidth_13	13	13
PTCPCLQRCodeUnitWidth_14	14	14
PTCPCLQRCodeUnitWidth_15	15	15
PTCPCLQRCodeUnitWidth_16	16	16
PTCPCLQRCodeUnitWidth_17	17	17
PTCPCLQRCodeUnitWidth_18	18	18
PTCPCLQRCodeUnitWidth_19	19	19
PTCPCLQRCodeUnitWidth_20	20	20
PTCPCLQRCodeUnitWidth_21	21	21
PTCPCLQRCodeUnitWidth_22	22	22
PTCPCLQRCodeUnitWidth_23	23	23
PTCPCLQRCodeUnitWidth_24	24	24
PTCPCLQRCodeUnitWidth_25	25	25
PTCPCLQRCodeUnitWidth_26	26	26

PTCPCLQRCodeUnitWidth_27	27	27
PTCPCLQRCodeUnitWidth_28	28	28
PTCPCLQRCodeUnitWidth_29	29	29
PTCPCLQRCodeUnitWidth_30	30	30
PTCPCLQRCodeUnitWidth_31	31	31
PTCPCLQRCodeUnitWidth_32	32	32

## 1.7 二维码纠错水平

- 代码

```
typedef NS_ENUM(NSUInteger, PTCPLQRCodeCorrectionLevel) {
    PTCPLQRCodeCorrectionLevelH = 0, ///< Ultra high reliability level
    PTCPLQRCodeCorrectionLevelQ = 1, ///< High reliability level
    PTCPLQRCodeCorrectionLevelM = 2, ///< Standard level
    PTCPLQRCodeCorrectionLevelL = 3 ///< High density level
};
```

- 说明

枚举成员	枚举值	描述
PTCPCLQRCodeCorrectionLevelH	0	极高可靠性级别
PTCPCLQRCodeCorrectionLevelQ	1	高可靠性级别
PTCPCLQRCodeCorrectionLevelM	2	标准级别
PTCPCLQRCodeCorrectionLevelL	3	高密度级别

## 1.8 二维码字符类型

- 代码

```
typedef NS_ENUM(NSUInteger, PTCPLQRCodeDataInputMode) {
    PTCPLQRCodeDataInputModeA = 0, ///< Automatic(default)
    PTCPLQRCodeDataInputModeM = 1 ///< manual
};
```

- 说明

枚举成员	枚举值	描述
PTCPCLQRCodeDataInputModeA	0	自动(默认)
PTCPCLQRCodeDataInputModeM	1	手动

## 1.9 文本的字体类型

- 代码

```
typedef NS_ENUM(NSUInteger, PTCPLTextFontName) {
    PTCPLTextFont1      = 1,    ///< chinese:24x24    english:9x17
    PTCPLTextFont2      = 2,    ///< chinese:24x24    english:8x16
    PTCPLTextFont3      = 3,    ///< chinese:20x20    english:10x20
    PTCPLTextFont4      = 4,    ///< chinese:32x32    english:16x32
    PTCPLTextFont5      = 5,    ///< chinese:24x24    english:12x24
    PTCPLTextFont7      = 7,    ///< chinese:24x24    english:12x24
    PTCPLTextFont8      = 8,    ///< chinese:24x24    english:12x24
    PTCPLTextFont20     = 20,   ///< chinese:16x16    english:8x16
    PTCPLTextFont28     = 28,   ///< chinese:28x28    english:14x28
    PTCPLTextFont55     = 55,   ///< chinese:16x16    english:8x16
};
```

- 说明

枚举成员	枚举值	描述
PTCPCLTextFont1	1	chinese:24x24 english:9x17
PTCPCLTextFont2	2	chinese:24x24 english:8x16
PTCPCLTextFont3	3	chinese:20x20 english:10x20
PTCPCLTextFont4	4	chinese:32x32 english:16x32
PTCPCLTextFont5	5	chinese:24x24 english:12x24
PTCPCLTextFont7	7	chinese:24x24 english:12x24
PTCPCLTextFont8	8	chinese:24x24 english:12x24
PTCPCLTextFont20	20	chinese:16x16 english:8x16
PTCPCLTextFont28	28	chinese:28x28 english:14x28
PTCPCLTextFont55	55	chinese:16x16 english:8x16

## 1.10 缩放文本字体名称

- 代码

```
typedef NS_ENUM(NSUInteger, PTCPCLScaledFontName) {
    PTCPCLScaledFontNameTTF      = 0,    ///< default
    PTCPCLScaledFontNameCSF      = 1,
    PTCPCLScaledFontNameFNT      = 2
};
```

- 说明

枚举成员	枚举值	描述
PTCPCLScaledFontNameTTF	0	含有TTF扩展名
PTCPCLScaledFontNameCSF	1	含有CSF扩展名
PTCPCLScaledFontNameFNT	2	含有FNT扩展名

## 1.11 字体组编号

- 代码

```
typedef NS_ENUM(NSUInteger, PTCPCLFontGroupNumber) {
    PTCPCLFontGroupNumber0      = 0,
    PTCPCLFontGroupNumber1      = 1,
    PTCPCLFontGroupNumber2      = 2,
    PTCPCLFontGroupNumber3      = 3,
    PTCPCLFontGroupNumber4      = 4,
    PTCPCLFontGroupNumber5      = 5,
    PTCPCLFontGroupNumber6      = 6,
    PTCPCLFontGroupNumber7      = 7,
    PTCPCLFontGroupNumber8      = 8,
    PTCPCLFontGroupNumber9      = 9,
    PTCPCLFontGroupNumber10     = 10
};
```

- 说明

枚举成员	枚举值	描述
PTCPCLFontGroupNumber0	0	字体组编号0
PTCPCLFontGroupNumber1	1	1
PTCPCLFontGroupNumber2	2	2
PTCPCLFontGroupNumber3	3	3
PTCPCLFontGroupNumber4	4	4
PTCPCLFontGroupNumber5	5	5
PTCPCLFontGroupNumber6	6	6
PTCPCLFontGroupNumber7	7	7
PTCPCLFontGroupNumber8	8	8
PTCPCLFontGroupNumber9	9	9
PTCPCLFontGroupNumber10	10	10

## 1.12 字体加粗

- 代码

```
typedef NS_ENUM(NSUInteger, PTCPClTextBold) {
    PTCPClTextBold0    = 0, ///< default, not bold
    PTCPClTextBold1    = 1,
    PTCPClTextBold2    = 2,
    PTCPClTextBold3    = 3,
    PTCPClTextBold4    = 4,
    PTCPClTextBold5    = 5
};
```

- 说明

枚举成员	枚举值	描述
PTCPCLTextBold0	0	默认值, 表示不加粗
PTCPCLTextBold1	1	加粗级别1
PTCPCLTextBold2	2	加粗级别2
PTCPCLTextBold3	3	加粗级别3
PTCPCLTextBold4	4	加粗级别4
PTCPCLTextBold5	5	加粗级别5

## 1.13 旋转角度

- 代码

```
typedef NS_ENUM(NSUInteger, PTCPClStyleRotation) {
    PTCPClStyleRotation0    = 0,    ///< Counterclockwise rotation, default
    PTCPClStyleRotation90   = 90,   ///< Counterclockwise rotation 90
    PTCPClStyleRotation180  = 180,  ///< Counterclockwise rotation 180
    PTCPClStyleRotation270  = 270   ///< Counterclockwise rotation 270
};
```

- 说明

枚举成员	枚举值	描述
PTCPCLStyleRotation0	0	默认值, 正常打印
PTCPCLStyleRotation90	90	旋转90°
PTCPCLStyleRotation180	180	旋转180°
PTCPCLStyleRotation270	270	旋转270°

## 1.14 字体放大系数

- 代码

```
typedef NS_ENUM(NSUInteger, PTCPCCLFontScale) {
    PTCPCCLFontScale_1 = 1,
    PTCPCCLFontScale_2,
    PTCPCCLFontScale_3,
    PTCPCCLFontScale_4,
    PTCPCCLFontScale_5,
    PTCPCCLFontScale_6,
    PTCPCCLFontScale_7,
    PTCPCCLFontScale_8,
    PTCPCCLFontScale_9,
    PTCPCCLFontScale_10,
    PTCPCCLFontScale_11,
    PTCPCCLFontScale_12,
    PTCPCCLFontScale_13,
    PTCPCCLFontScale_14,
    PTCPCCLFontScale_15,
    PTCPCCLFontScale_16
};
```

- 说明

枚举成员	枚举值	描述
PTCPCLFontScale_1	1	放大1倍
PTCPCLFontScale_2	2	2
PTCPCLFontScale_3	3	3
PTCPCLFontScale_4	4	4
PTCPCLFontScale_5	5	5
PTCPCLFontScale_6	6	6
PTCPCLFontScale_7	7	7
PTCPCLFontScale_8	8	8
PTCPCLFontScale_9	9	9
PTCPCLFontScale_10	10	10
PTCPCLFontScale_11	11	11
PTCPCLFontScale_12	12	12
PTCPCLFontScale_13	13	13
PTCPCLFontScale_14	14	14
PTCPCLFontScale_15	15	15
PTCPCLFontScale_16	16	16

## 1.15 泰文字体类型

- 代码

```
typedef NS_ENUM(NSUInteger, PTCPLThaiFontStyle) {
    PTCPLThaiFontStyle0 = 0,    ///< big font
    PTCPLThaiFontStyle1 = 1    ///< small font
};
```

- 说明

枚举成员	枚举值	描述
PTCPCLThaiFontStyle0	0	泰文大字体
PTCPCLThaiFontStyle1	1	泰文小字体

## 1.16 PDF417码的纠错等级

- 代码

```
typedef NS_ENUM(NSUInteger, PTCPLPDF417ErrLevel) {
    PTCPLPDF417ErrLevel_0 = 0,
    PTCPLPDF417ErrLevel_1 = 1,    //default
    PTCPLPDF417ErrLevel_2,
    PTCPLPDF417ErrLevel_3,
    PTCPLPDF417ErrLevel_4,
    PTCPLPDF417ErrLevel_5,
    PTCPLPDF417ErrLevel_6,
    PTCPLPDF417ErrLevel_7,
    PTCPLPDF417ErrLevel_8
};
```

- 说明

枚举成员	枚举值	描述
PTCPCLPDF417ErrLevel_0	0	错误恢复级别0
PTCPCLPDF417ErrLevel_1	1	1
PTCPCLPDF417ErrLevel_2	2	2
PTCPCLPDF417ErrLevel_3	3	3
PTCPCLPDF417ErrLevel_4	4	4
PTCPCLPDF417ErrLevel_5	5	5
PTCPCLPDF417ErrLevel_6	6	6
PTCPCLPDF417ErrLevel_7	7	7
PTCPCLPDF417ErrLevel_8	8	8

## 1.17 PDF417码的方向

- 代码

```
typedef NS_ENUM(NSUInteger, PTCPLPDF417Direction) {  
    PTCPLPDF417DirectionH    = 0,    ///< horizontal  
    PTCPLPDF417DirectionV    = 1    ///< vertical  
};
```

- 说明

枚举成员	枚举值	描述
PTCPLPDF417DirectionH	0	水平方向
PTCPLPDF417DirectionV	1	垂直方向

## 1.18 PDF417数据压缩模式

- 代码

```
typedef NS_ENUM(NSUInteger, PTCPLPDF417Binary) {  
    PTCPLPDF417BinaryA    = 0,    ///< automatic  
    PTCPLPDF417BinaryF    = 1    ///< forced  
};
```

- 说明

枚举成员	枚举值	描述
PTCPLPDF417BinaryA	0	
PTCPLPDF417BinaryF	1	

## 2.属性

---

### 2.1 cmdData

- 代码

```
@property(strong, nonatomic, readwrite) NSMutableData * _Nonnull cmdData;
```

- 说明

这个是最终下发给打印机的数据

## 2.2 encoding

- 代码

```
@property (nonatomic, assign) NSStringEncoding encoding;
```

- 说明

指令的编码格式，默认是GBK的编码

NSStringEncoding类型，其他的编码格式可在苹果CoreFoundation框架下的CFStringEncodingExt.h类中查看

## 3.接口

### 3.1 添加指令文本

- 描述

该接口用于自己封装的指令添加到cmdData数据中进行下发

- 代码

```
- (void)appendCommand:(NSString * _Nonnull)cmd;
```

- 参数

参数	类型	描述	范围
cmd	NSString	封装好的字符串指令	符合标准的CPCL指令

### 3.2 添加指令数据

- 描述

该接口用于自己生成的data数据添加到cmdData数据中进行下发

- 代码

```
- (void)appendCommandData:(NSData * _Nonnull)data;
```

- 参数

参数	类型	描述
data	NSData	自己生成的data数据

### 3.3 打印自检页

- 描述

下发该指令后，打印机会打印本机的基本信息

- 代码

```
- (void)printSelfInspectionPage;
```

### 3.4 获取打印机状态

- 描述

用于获取打印机目前的状态

- 代码

```
- (void)cpclGetPaperStatus;
```

- 返回值

- 返回值是一个字节的数据，通过 `- (void)whenReceiveData:(PTDataParameterBlock)receiveDataBlock;` 这个接口可获取，需要对这个字节进行位处理

位	描述
bit0	0:空闲 1:打印中
bit1	0:有纸 1:缺纸
bit2	0:合盖 1:开盖
bit3	0:电量充足 1:低电量
bit4	保留
bit5	保留
bit6	保留
bit7	保留

eg:

空闲: 00

缺纸: 02

走纸或打印中: 01

开盖有纸: 04

开盖缺纸: 06

### 3.5 打印完成状态的开关

- 描述

- 用于获取数据打印完成后打印机的状态，目前有三种状态：打印成功、打印失败(开盖)、打印失败(缺纸)
- 打印机固件需要支持该指令
- 该接口的返回值用接口 `- (void)whenUpdatePrintState: (PTPrintStateBlock)printStateBlock;` 处理

- 代码

```
- (void)cpc1TurnOnPrintStatusCallBack:(BOOL)flag;
```

- 参数

参数	类型	描述
flag	BOOL	YES:开启打印状态 NO:关闭打印状态

- 返回值

```
/*!
 * \~chinese
 * 接收到打印机打印状态回调，针对CPCL ESC指令
 *
 * @param printStateBlock 回调block
 *
 * \~english
 * Trigger this method when receiving print state ,For CPCL and ESC
 instructions
 *
 * @param printStateBlock Trigger block
 */
- (void)whenUpdatePrintState:(PTPrintStateBlock)printStateBlock;
```

### 3.6 打开下划线

- 描述

打开下划线

- 代码

```
- (void)cpclUnderlineON;
```

### 3.7 关闭下划线

- 描述

关闭下划线功能

- 代码

```
- (void)cpclUnderlineOFF;
```

### 3.8 获取SN号

- 描述

获取打印机的序列号

- 代码

```
- (void)cpc1GetPrinterSN;
```

- 返回值

- 返回的是一个以5f开头，00结尾的十六进制数据
- 去掉头尾后，在 `PTEncode` 类中可用 `+(NSString *)decodeStringWithData:(NSData *)data;` 解析

```
/*!  
 *  \~chinese  
 *  解码,default:GBK  
 *  
 *  @param data          需要解码的数据  
 *  
 *  @return 解码后的字符串  
 *  
 *  \~english  
 *  Decoding,default:GBK  
 *  
 *  @param data          Data to be decoded  
 *  
 *  @return Decoded data  
 */  
+(NSString *)decodeStringWithData:(NSData *)data;
```

### 3.9 设置纸张类型

- 描述

用于设置打印纸张的类型

- 代码

```
- (void)cpc1PaperTypeWithType:(PTCPLPaperType)type;
```

- 参数

参数	类型	描述	范围
type	PTCPCLPaperType	设置的纸张类型	查看上面枚举

### 3.10 初始化标签

- 描述

- 页模式打印最开始就要调用该接口
- 页模式打印时，调用完该接口后，才可其他的接口
- 可设置标签份数、高度、分辨率

- 代码

```
- (void)cpcLabelWithOffset:(NSInteger)offset
    hRes:(PTCPCLLabelResolution)hRes
    vRes:(PTCPCLLabelResolution)vRes
    height:(NSInteger)height
    quantity:(NSInteger)quantity;
```

- 参数

参数	类型	描述	范围
offset	NSInteger	偏移量,以点为单位	0-65535
hRes	PTCPCLLabelResolution	水平分辨率	查看上面枚举
vRes	PTCPCLLabelResolution	垂直分辨率	查看上面枚举
height	NSInteger	标签高度,以点为单位	0-65535
quantity	NSInteger	打印的份数	0-1024

### 3.11 指定标签宽度

- 描述

- 设置打印标签的宽度
- 一般初始化标签后再调用该接口，用于设置标签宽度

- 代码

```
- (void)cpc1Pagewidth:(NSInteger)pagewidth;
```

- 参数

参数	类型	描述	范围
pageWidth	NSInteger	标签宽度, 以点为单位	0-65535

## 3.12 一维条码

### 3.12.1 横向条码

- 描述

- 打印水平条码
- 每种条码对应的数据格式可能不同, 打印条码时应注意

- 代码

```
- (void)cpc1Barcode:(PTCPCLBarcodeStyle)type  
    width:(NSInteger)width  
    ratio:(PTCPCLBarcodeBarRatio)ratio  
    height:(NSInteger)height  
    x:(NSInteger)x  
    y:(NSInteger)y  
    barcode:(NSString * _Nonnull)barcode;
```

- 参数

参数	类型	描述	范围
type	PTCPCLBarcodeStyle	条码类型	查看上面枚举
width	NSInteger	窄条宽度	0 to 65535
ratio	PTCPCLBarcodeBarRatio	款窄条比例	查看上面枚举
height	NSInteger	条码高度	0 to 65535
x	NSInteger	横向起始位置	0 to 65535
y	NSInteger	纵向起始位置	0 to 65535
barcode	NSString	条码内容	Up to 8191 bytes of Alpha Numeric Data.

### 3.12.2 纵向条码

- 描述

- 打印纵向条码
- 每种条码对应的数据格式可能不同，打印条码时应注意

- 代码

```
- (void)cpc1BarcodeVertical:(PTCPCLBarcodeStyle)type
    width:(NSInteger)width
    ratio:(PTCPCLBarcodeBarRatio)ratio
    height:(NSInteger)height
    x:(NSInteger)x
    y:(NSInteger)y
    barcode:(NSString * _Nonnull)barcode;
```

- 参数

参数	类型	描述	范围
type	PTCPCLBarcodeStyle	条码类型	查看上面枚举
width	NSInteger	窄条宽度	0 to 65535
ratio	PTCPCLBarcodeBarRatio	宽窄条比例	查看上面枚举
height	NSInteger	条码高度	0 to 65535
x	NSInteger	横向起始位置	0 to 65535
y	NSInteger	纵向起始位置	0 to 65535
barcode	NSString	条码内容	Up to 8191 bytes of Alpha Numeric Data.

### 3.12.3 条码下方文字

- 描述

- 该接口有两条，效果是一样的，参数不同
- 这项命令避免了使用单独文本命令注释条码的必要
- 若不调用该接口，也可以通过文本接口显示条码内容,只是更麻烦
- 该条指令需要和结束条码展示的接口一起使用

- 代码

```
- (void)cpc1BarcodeTextWithFont:(PTCPCLTextFontName) font
    fontSize:(NSInteger)fontSize
    offset:(NSInteger)offset;
```

```
- (void)cpc1BarcodeTextWithTrueTypeFont:(PTCPCLTextFontName) font
    xScale:(NSInteger)xScale
    yScale:(NSInteger)yScale
    offset:(NSInteger)offset;
```

- 参数

参数	类型	描述	范围
font	PTCPCLTextFontName	字体类型	查看上面枚举
fontSize	NSInteger	字体大小	0-999
offset	NSInteger	文本距离条码的偏移量	0-999

参数	类型	描述	范围
font	PTCPCLTextFontName	字体类型	查看上面枚举
xScale	NSInteger	字符横向大小	0-999
yScale	NSInteger	字符纵向大小	0-999
offset	NSInteger	文本距离条码的偏移量	0-999

### 3.12.4 结束条码底部文字展示

- 描述

- 结束条码下方的文字
- 该条指令和条码下方文字接口一起使用

- 代码

```
- (void)cpc1BarcodeTextOff;
```

## 3.13 二维码

### 3.13.1 横向QR码

- 描述

- 打印横向二维码
- 该接口应和添加条码数据、结束QR码接口一起使用

- 代码

```
- (void)cpc1BarcodeQRcodewithXPos:(NSInteger)xPos
    yPos:(NSInteger)yPos
    model:(PTCPCLQRCodeModel)model
    unitwidth:(PTCPCLQRCodeUnitwidth)unitwidth;
```

- 参数

参数	类型	描述	范围
xPos	NSInteger	横向开始点	0 to 65535
yPos	NSInteger	纵向开始点	0 to 65535
model	PTCPCLQRCodeModel	规范编号	查看上面枚举
unitWidth	PTCPCLQRCodeUnitWidth	单元宽度	查看上面枚举

### 3.13.2 纵向QR码

- 描述

- 打印纵向二维码
- 该接口应和添加条码数据、结束QR码接口一起使用

- 代码

```
- (void)cpc1BarcodeVerticalQRcodewithXPos:(NSInteger)xPos
    yPos:(NSInteger)yPos
    model:(PTCPCLQRCodeModel)model
    unitwidth:(PTCPCLQRCodeUnitwidth)unitwidth;
```

- 参数

参数	类型	描述	范围
xPos	NSInteger	横向开始点	0 to 65535
yPos	NSInteger	纵向开始点	0 to 65535
model	PTCPCLQRCodeModel	规范编号	查看上面枚举
unitWidth	PTCPCLQRCodeUnitWidth	单元宽度	查看上面枚举

### 3.13.3 添加QR码数据

- 描述

- 添加二维码数据
- 配合上面的接口使用

- 代码

```
- (void)cpc1BarcodeQRCodeCorrectionLevel:(PTCPCLQRCodeCorrectionLevel)level  
characterMode:(PTCPCLQRCodeDataInputMode)characterMode context:(NSString  
*_Nonnull)context;
```

- 参数

参数	类型	描述	范围
level	PTCPCLQRCodeCorrectionLevel	纠错级别	查看上面枚举
characterMode	PTCPCLQRCodeDataInputMode	输入的字符模式符号	查看上面枚举
context	NSString	二维码数据	

### 3.13.4 结束QR码

- 描述

表示QR码的结束

- 代码

```
- (void)cpc1BarcodeQRcodeEnd;
```

### 3.14 PDF417码

- 描述

- 打印PDF417要用到三条指令，配置参数、添加数据、结束PDF417码
- 该命令有许多可选参数，可以指定任意数量的参数。如果多次指定参数，则使用指定的最后一个值。

- 代码

```

- (void)cpc1PDF417CodeWithDirection:(PTCPLPDF417Direction)directon
    xPos:(NSInteger)xPos
    yPos:(NSInteger)yPos
    xDot:(NSInteger)xDot
    yDot:(NSInteger)yDot
    columns:(NSInteger)columns
    rows:(NSInteger)rows
    ecc:(PTCPLPDF417ErrLevel)ecc
    binaryModel:(PTCPLPDF417Binary)binaryModel;

- (void)cpc1PDF417CodeAddContext:(NSString *)context;

- (void)cpc1PDF417codeEnd;

```

- 参数

参数	类型	描述	范围
directon	PTCPLPDF417Direction	水平或者垂直	查看上面枚举
xPos	NSInteger	X起始位置	0 to 65535
yPos	NSInteger	Y起始位置	0 to 65535
xDot	NSInteger	单个元素的X大小（以点为单位），默认是2	0 to 65535
yDot	NSInteger	单个元素的Y大小（以点为单位），默认是6	0 to 65535
columns	NSInteger	指定要使用的列数 (1 to 30), 默认是3	1 to 30
rows	NSInteger	指定要使用的行数 (0 to 90), 默认是0	0 to 90
ecc	PTCPLPDF417ErrLevel	指定错误恢复级别 (0 to 8), 默认是1	查看上面枚举
binaryModel	PTCPLPDF417Binary	强制二进制压缩模式 (0 or 1), 默认是0	查看上面枚举

### 3.15 绘制矩形

- 描述

- 用于绘制框
- 默认情况下，接口以纯黑色绘制一个框。
- 该接口可以与对齐接口一起使用以对齐框。

- 代码

```
- (void)cpc1BoxwithXPos:(NSInteger)xPos  
    yPos:(NSInteger)yPos  
    xEnd:(NSInteger)xEnd  
    yEnd:(NSInteger)yEnd  
    thickness:(NSInteger)thickness;
```

- 参数

参数	类型	描述	范围
xPos	NSInteger	横向开始点	0~65535
yPos	NSInteger	纵向开始点	0~65535
xEnd	NSInteger	横向结束点	0~65535
yEnd	NSInteger	纵向结束点	0~65535
thickness	NSInteger	线条宽度	0~65535

## 3.16 绘制线条

- 描述

- 用于绘制线条。
- 具有两种操作模式，基于线的一侧是否是平坦的。

- 代码

```
- (void)cpc1LinewithXPos:(NSInteger)xPos  
    yPos:(NSInteger)yPos  
    xEnd:(NSInteger)xEnd  
    yEnd:(NSInteger)yEnd  
    thickness:(NSInteger)thickness;
```

- 参数

参数	类型	描述	范围
xPos	NSInteger	起点的x坐标	0~65535
yPos	NSInteger	起点的y坐标	0~65535
xEnd	NSInteger	终点的x坐标	0~65535
yEnd	NSInteger	终点的y坐标	0~65535
thickness	NSInteger	线条的单位宽度	0~65535

### 3.17 绘制反白线条

- 描述

- 用于绘制一条线，该线反转绘制它的标签区域。
- 具有两种操作模式，基于线的一侧是否是平坦的。

- 代码

```
- (void)cpclInverseLinewithXPos:(NSInteger)xPos
    yPos:(NSInteger)yPos
    xEnd:(NSInteger)xEnd
    yEnd:(NSInteger)yEnd
    thickness:(NSInteger)thickness;
```

- 参数

参数	类型	描述	范围
xPos	NSInteger	起点的x坐标	0~65535
yPos	NSInteger	起点的y坐标	0~65535
xEnd	NSInteger	终点的x坐标	0~65535
yEnd	NSInteger	终点的y坐标	0~65535
thickness	NSInteger	线条的单位宽度	0~65535

### 3.18 打印图片

- 描述

- 用于将原始二进制位图数据打印到标签

- 代码

```
- (void)cpc1PrintBitmapWithXPos:(NSInteger)xPos  
    yPos:(NSInteger)yPos  
    image:(CGImageRef)image  
    bitmapMode:(PTBitmapMode)bitmapMode  
    compress:(PTBitmapCompressMode)compress;
```

- 参数

参数	类型	描述	范围
xPos	NSInteger	横向开始打印位置	0~65535
yPos	NSInteger	纵向开始打印位置	0~65535
image	CGImageRef	需要打印的图片	图片
bitmapMode	PTBitmapMode	位图的类型，黑白和灰阶抖动	在开放类PTBitmap查看
compress	PTBitmapCompressMode	打印机支持的压缩类型	在开放类PTBitmap查看

## 3.19 串联文本

- 描述

- 拼接的文本高度不能超过标签高度
- 打印串联文本需要多个接口结合使用，先开始串联文本，然后拼接文本(可缩放的文本、字体组文本)、结束串联文本

- 代码

- 开始横向串联文本

```
- (void)cpc1ConcatStartWithXPos:(NSInteger)xPos yPos:(NSInteger)yPos;
```

- 开始纵向串联文本

```
- (void)cpc1ConcatVerticalStartWithXPos:(NSInteger)xPos yPos:  
(NSInteger)yPos;
```

- 拼接文本

```
- (void)cpc1ConcatTextWithFont:(PTCPCLTextFontName)font
    fontSize:(NSInteger)fontSize
    offset:(NSInteger)offset
    text:(NSString * _Nonnull)text;
```

- 拼接可缩放的水平文本

```
- (void)cpc1ConcatScaleTextWithScaledFont:(NSInteger)scaledFont
    xScale:(NSInteger)xScale
    yScale:(NSInteger)yScale
    offset:(NSInteger)offset
    text:(NSString * _Nonnull)text;
```

- 拼接可缩放的垂直文本

```
- (void)cpc1ConcatVerticalScaleTextWithScaledFont:(NSInteger)scaledFont
    xScale:(NSInteger)xScale
    yScale:(NSInteger)yScale
    offset:(NSInteger)offset
    text:(NSString *
_Nonnull)text;
```

- 拼接字体组文本

```
- (void)cpc1ConcatTextWithFontGroup:(PTCPCLFontGroupNumber)fontGroup
    offset:(NSInteger)offset
    text:(NSString * _Nonnull)text;
```

- 结束串联文本

```
- (void)cpc1ConcatEnd;
```

- 参数

- 开始横向串联文本

参数	类型	描述	范围
xPos	NSInteger	横向起始位置	0~65535
yPos	NSInteger	纵向起始位置	0~65535

- 开始纵向串联文本

参数	类型	描述	范围
xPos	NSInteger	横向起始位置	0~65535
yPos	NSInteger	纵向起始位置	0~65535

- 拼接文本

参数	类型	描述	范围
font	PTCPCLTextFontName	字体类型	查看上面枚举
fontSize	NSInteger	字体大小	0~65535
offset	NSInteger	文字纵向偏移(以开始文字串联Y的位置为标准)	0~65535
text	NSString	文本内容	Up to 2024 characters

- 拼接可缩放的水平文本\拼接可缩放的垂直文本

参数	类型	描述	范围
scaledFont	NSInteger	可缩放字体	0~65535
xScale	NSInteger	字体横向大小(点).	0~65535
yScale	NSInteger	字体纵向大小(点)	0~65535
offset	NSInteger	文字纵向偏移(以开始文字串联Y的位置为标准)	0~65535
text	NSString	文本内容	Up to 2024 characters

- 拼接字体组文本

参数	类型	描述	范围
fontGroup	PTCPCLFontGroupNumber	字体组编码(0-10)	查看上面枚举
offset	NSInteger	文字纵向偏移(以开始文字串联Y的位置为标准)	0~65535
text	NSString	文本内容	Up to 2024 characters

## 3.20 文本

### 3.20.1 打印文本

- 描述

- 用于打印正常的文本
- 如果文本过长时应使用自动换行文本接口
- 可结合对齐方式指令使用

- 代码

```
- (void)cpclTextWithRotate:(PTCPCLStyleRotation)rotate
    font:(PTCPCLTextFontName)font
    fontSize:(NSInteger)fontSize
    x:(NSInteger)x
    y:(NSInteger)y
    text:(NSString * _Nonnull)text;
```

- 参数

参数	类型	描述	范围
rotate	PTCPCLStyleRotation	文字旋转方向	查看上面枚举
font	PTCPCLTextFontName	字体类型	查看上面枚举
fontSize	NSInteger	字体大小	0~65535
x	NSInteger	起始位置x	0~65535
y	NSInteger	起始位置y	0~65535
text	NSString	打印内容	Up to 8191 characters

### 3.20.2 反白加粗文本

- 描述

和打印文本的接口对比加了反白、加粗的参数

- 代码

```
- (void)appendTextWithFont:(PTCPCLTextFontName)font
    rotate:(PTCPCLStyleRotation)rotate
    fontSize:(NSInteger)fontSize
    reverse:(BOOL)reverse
    bold:(PTCPCLTextBold)bold
    x:(NSInteger)x
    y:(NSInteger)y
    text:(NSString * _Nonnull)text;
```

- 参数

参数	类型	描述	范围
font	PTCPCLTextFontName	字体类型	查看上面枚举
rotate	PTCPCLStyleRotation	文字旋转方向	查看上面枚举
fontSize	NSInteger	字体大小	0~65535
reverse	BOOL	是否反白	Yes or no
bold	PTCPCLTextBold	加粗	查看上面枚举
x	NSInteger	起始位置x	0~65535
y	NSInteger	起始位置y	0~65535
text	NSString	打印内容	Up to 8191 characters

### 3.20.3 居中文本

- 描述

- 用于打印居中文本
- 如果旋转90°或者270°，该条指令无效
- 打印区域宽度内居中效果

- 代码

```
- (void)cpc1CenterTextWithRotate:(PTCPCLStyleRotation)rotate
    font:(PTCPCLTextFontName)font
    fontSize:(NSInteger)fontSize
    x:(NSInteger)x
    y:(NSInteger)y
    width:(NSInteger)width
    text:(NSString * _Nonnull)text;
```

- 参数

参数	类型	描述	范围
rotate	PTCPCLStyleRotation	文字旋转方向	查看上面枚举
font	PTCPCLTextFontName	字体类型	查看上面枚举
fontSize	NSInteger	字体大小	0~65535
x	NSInteger	起始位置x	0~65535
y	NSInteger	起始位置y	0~65535
width	NSInteger	打印区域宽度	0~65535
text	NSString	打印内容	Up to 8191 characters

### 3.20.4 自动换行文本

- 描述

- 用于解决文本长度超过打印区域宽度时，文本能自动换行打印
- 超过打印区域高度部分将不会打印

- 代码

```
- (NSInteger)cpc1AutoTextWithRotate:(PTCPCLStyleRotation)rotate
    font:(PTCPCLTextFontName)font
    fontSize:(NSInteger)fontSize
    x:(NSInteger)x
    y:(NSInteger)y
    safeHeight:(NSInteger)safeHeight
    width:(NSInteger)width
    lineSpacing:(NSInteger)lineSpacing
    text:(NSString * _Nonnull)text;
```

- 参数

参数	类型	描述	范围
rotate	PTCPCLStyleRotation	文字旋转方向	查看上面枚举
font	PTCPCLTextFontName	字体类型	查看上面枚举
fontSize	NSInteger	字体大小	0~65535
x	NSInteger	起始位置x	0~65535
y	NSInteger	起始位置y	0~65535
safeHeight	NSInteger	打印区域高度,超过高度部分不会打印	0~65535
width	NSInteger	打印区域宽度,宽度为0表示不自动换行	0~65535
lineSpacing	NSInteger	行间隔	0~65535
text	NSString	打印内容	Up to 8191 characters

- 返回值

类型	描述	范围
NSInteger	自动换行后需要的高度	0~65535

### 3.20.5 自动换行居中文本

- 描述

- 自动换行的基础上增加居中的效果
- 解决旋转90°或者270°后,居中对齐方式接口无效果,则需要用到此接口

- 代码

```
- (NSInteger)cpc1AutoTextWithRotate:(PTCPCLStyleRotation)rotate
    font:(PTCPCLTextFontName)font
    fontSize:(NSInteger)fontSize
    xPos:(NSInteger)xPos
    yPos:(NSInteger)yPos
    center:(BOOL)center
    safeHeight:(NSInteger)safeHeight
    width:(NSInteger)width
    lineSpacing:(NSInteger)lineSpacing
    text:(NSString * _Nonnull)text
```

- 参数

参数	类型	描述	范围
rotate	PTCPCLStyleRotation	文字旋转方向	查看上面枚举
font	PTCPCLTextFontName	字体类型	查看上面枚举
fontSize	NSInteger	字体大小	0~65535
xPos	NSInteger	起始位置x	0~65535
yPos	NSInteger	起始位置y	0~65535
center	BOOL	是否居中	YES or NO
safeHeight	NSInteger	打印区域高度,超过高度部分不会打印	0~65535
width	NSInteger	打印区域宽度,宽度为0表示不自动换行	0~65535
lineSpacing	NSInteger	行间隔	0~65535
text	NSString	打印内容	Up to 8191 characters

- 返回值

类型	描述	范围
NSInteger	自动换行后需要的高度	0~65535

### 3.20.6 泰文自动换行

- 描述

- 用于泰文自动换行打印

- 代码

```
- (void)cpc1AutoThaiTextWithFont:(PTCPCLThaiFontStyle) font
    fontSize:(NSInteger)fontSize
    x:(NSInteger)x
    y:(NSInteger)y
    lineSpace:(NSInteger)lineSpace
    width:(NSInteger)width
    text:(NSString * _Nonnull)text;
```

- 参数

参数	类型	描述	范围
font	PTCPCLThaiFontStyle	字体类型	查看上面枚举
fontSize	NSInteger	字体大小	0~65535
x	NSInteger	起始位置x	0~65535
y	NSInteger	起始位置y	0~65535
lineSpacing	NSInteger	行间隔,默认30	0~65535
text	NSString	打印内容	Up to 8191 characters

## 3.21 对齐方式

### 3.21.1 左对齐

- 描述

- 用于更改支持字段的对齐方式
- 默认是左对齐

- 代码

```
- (void)cpc1Left;
- (void)cpc1Left:(NSInteger)range;
```

- 参数

参数	类型	描述	范围
range	NSInteger	用于为FONT-GROUP指令指定文字区域	0 to 65535

### 3.21.2 居中

- 描述

- 用于更改支持字段的对齐方式
- 居中对齐，断电后不会保存

- 代码

```
- (void)cpc1Center;  
- (void)cpc1CenterWithRange:(NSInteger)range;
```

- 参数

参数	类型	描述	范围
range	NSInteger	居中区域宽度	0 to 65535

### 3.21.3 右对齐

- 描述

- 用于更改支持字段的对齐方式
- 右对齐对齐，断电后不会保存

- 代码

```
- (void)cpc1Right;  
- (void)cpc1Right:(NSInteger)right;
```

- 参数

参数	类型	描述	范围
range	NSInteger	对齐的右边界位置	0 to 65535

## 3.22 打印标签

- 描述

- 用于终止CPCL标签会话，并创建结果打印输出
- 必须使用该接口终止每个CPCL标签会话

- 代码

```
- (void)cpc1Print;
```

### 3.23 反向打印标签

- 描述

- 用于终止CPCL标签会话，并创建结果旋转180°后打印输出

- 代码

```
- (void)cpc1PoPrint;
```

### 3.24 多行文本

- 描述

- 用于使用相同的字体打印多行文本，而无需手动指定每行的间距或位置。
- 该接口先调用开始多行文本打印接口，后面调用文本的接口，最后结束多行打印接口

- 代码

```
//开始多行文本打印。  
- (void)cpc1MultiLineStartWithLineHeight:(NSInteger)lineHeight;  
  
//中间需要调用文本打印的接口  
  
//结束多行文本打印。  
- (void)cpc1MultiLineEnd;
```

- 参数

参数	类型	描述	范围
lineHeight	NSInteger	每行的高度(点)	0 to 65535

## 3.25 缩放字体文本

- 描述
  - 用于打印缩放文本
  - 水平缩放文本和垂直缩放文本
- 代码

```
//水平打印缩放文本或TrueType字体文字
- (void)cpc1ScaleText:(PTCPLScaledFontName)scaledFont
    xScale:(NSInteger)xScale
    yScale:(NSInteger)yScale
    x:(NSInteger)x
    y:(NSInteger)y
    text:(NSString * _Nonnull)text;

//纵向打印缩放文本或TrueType字体文字
- (void)cpc1ScaleTextVertical:(PTCPLScaledFontName)scaledFont
    xScale:(NSInteger)xScale
    yScale:(NSInteger)yScale
    x:(NSInteger)x
    y:(NSInteger)y
    text:(NSString * _Nonnull)text;
```

- 参数

参数	类型	描述	范围
scaledFont	PTCPLScaledFontName	字体名	查看上面枚举
xScale	NSInteger	横向字体大小(点)	0 to 65535
yScale	NSInteger	纵向字体大小(点)	0 to 65535
x	NSInteger	横向起始位置	0 to 65535
y	NSInteger	纵向起始位置	0 to 65535
text	NSString	打印内容	Up to 8191 characters

## 3.26 字体加粗

- 描述

- 用于对文本字体加粗
- 需要在调用文本之前设置该接口

- 代码

```
- (void)cpc1SetBold:(PTCPCLTextBold)boldness;
```

- 参数

参数	类型	描述	范围
boldness	PTCPCLTextBold	加粗程度	查看上面枚举

## 3.27 字间距

- 描述

- 用于设置文本字体间隔
- 需要在调用文本之前设置该接口

- 代码

```
- (void)cpc1SetSpacing:(NSInteger)spacing;
```

- 参数

参数	类型	描述	范围
spacing	NSInteger	字间距	0 to 65535

## 3.28 字体放大倍数

- 描述

- 将常驻字体放大指定的放大倍数
- 需要在调用文本前设置该接口

- 代码

```
- (void)cpc1SetMagwithwidth:(PTCPCLFontScale)width height:
(PTCPCLFontScale)height;
```

- 参数

参数	类型	描述	范围
width	PTCPCLFontScale	宽度放大系数	查看上面枚举
height	PTCPCLFontScale	高度放大系数	查看上面枚举

## 3.29 水印文本

- 描述

- 先设置背景黑度值，再设置背景文字

- 代码

```
/*!
 * \~chinese
 *
 * 设置背景黑度值.(0-255) 0: normal
 *
 * \~english
 *
 */
- (void)cpc1SetBackgroundBlacknessValue:(NSInteger)value;

/*!
 * \~chinese
 *
 * 设置背景文字.
 *
 * \~english
 *
 * set back text.
 *
 */
- (void)cpc1BackTextWithFont:(PTCPCLTextFontName)font
    fontSize:(NSInteger)fontSize
    xPos:(NSInteger)xPos
    yPos:(NSInteger)yPos
    text:(NSString *_Nonnull)text;
```

- 参数
  - 背景黑度值

参数	类型	描述	范围
value	NSInteger	背景黑度值,0是正常	0-255

- 背景文字

参数	类型	描述	范围
font	PTCPCLTextFontName	字体类型	查看上面枚举
fontSize	NSInteger	字体大小	0 to 65535
xPos	NSInteger	起始位置xPos	0 to 65535
yPos	NSInteger	起始位置yPos	0 to 65535
text	NSString	打印内容	Up to 8191 characters

### 3.30 左边距

- 描述
  - 行打印模式下设置左边距
  - 将所有内容打印偏移设置的大小

- 代码

```
- (void)cpc1LineMargin:(NSInteger)offset;
```

- 参数

参数	类型	描述	范围
offset	NSInteger	距离左边的偏移量	0 to 65535

## 3.31 走纸

- 描述

- 设置走纸一行
- 设置走纸距离

- 代码

```
/*!
 *  \~chinese
 *
 *  走纸一行
 *
 *  \~english
 *
 *   feed
 *
 */
- (void)cpc1LineFeed;

/*!
 *  \~chinese
 *
 *  设置走纸
 *
 *  @param amount  走纸多少个点. -4000~4000dots
 *
 *  \~english
 *
 *  Set feed
 *
 *  @param amount  How much to feed in units.-4000~4000dots
 *
 */
- (void)cpc1Feed:(NSInteger)amount;
```

- 参数

参数	类型	描述	范围
amount	NSInteger	走纸多少个点	-4000~4000

## 3.32 打印速度

- 描述

用于设置打印速度

- 代码

```
- (void)cpc1Speed:(NSInteger)value;
```

- 参数

参数	类型	描述	范围
value	NSInteger	打印速度	0-5

## 3.33 定位缝标

- 描述

- 用于缝标的定位，打印结束后自动定位
- 用于标签纸，在调用打印接口前面调用该接口

- 代码

```
- (void)cpc1Form;
```

## 3.34 打印浓度

- 描述

用于设置打印浓度

- 代码

```
- (void)cpc1Tone:(NSInteger)value;
```

- 参数

参数	类型	描述	范围
value	NSInteger	设置打印浓度,默认是0	-100~200

### 3.35 行模式开启

- 描述

行模式打印数据时需要先调用该接口

- 代码

```
- (void)cpc1UtilitySession;
```

### 3.36 行模式指令前缀

- 描述

- 行模式打印时需要先调用该接口
- 这个接口和上面接口类似

- 代码

```
- (void)cpc1LineMode;
```

### 3.37 关闭倒带

- 描述

用于关闭倒带，打开电源时默认是开启的

- 代码

```
- (void)cpc1RewindOFF;
```

### 3.38 设置字符代码页

- 描述

设置打印机的字符代码页，设置后才可打印对应的字体

- 代码

```
/*!
 * \~chinese
 *
 * 设置字符代码页.
 *
 * @param codepage "USA" "FRANCE" "GERMANY" "UK" "DENMARK" "SWEDEN" "ITALY"
"SPAIN" "JAPAN-S" "NORWAY" "DENMARK II" "SPAIN II" "LATIN9" "KOREA"
"SLOVENIA" "CHINA" "BIG5" "CP874" "CP850" "CP437" "CP860" "CP863" "CP865"
"CP866" "CP852" "CP858" "CP857" "CP737" "CP720" "CP775" "CP855" "CP862"
"CP864" "ISO8859-6" "ISO8859-8" "ISO8859-9" "ISO8859-15" "WPC1252" "WPC1250"
"WPC1251" "WPC1252" "WPC1254" "WPC1255" "WPC1256" "ISO8859-1" "ISO8859-2"
"ISO8859-3" "ISO8859-4" "ISO8859-5" "TIS11" "TIS18"
 *
 * \~english
 *
 * Set Character CodePage
 *
 * @param codepage "USA" "FRANCE" "GERMANY" "UK" "DENMARK" "SWEDEN" "ITALY"
"SPAIN" "JAPAN-S" "NORWAY" "DENMARK II" "SPAIN II" "LATIN9" "KOREA"
"SLOVENIA" "CHINA" "BIG5" "CP874" "CP850" "CP437" "CP860" "CP863" "CP865"
"CP866" "CP852" "CP858" "CP857" "CP737" "CP720" "CP775" "CP855" "CP862"
"CP864" "ISO8859-6" "ISO8859-8" "ISO8859-9" "ISO8859-15" "WPC1252" "WPC1250"
"WPC1251" "WPC1252" "WPC1254" "WPC1255" "WPC1256" "ISO8859-1" "ISO8859-2"
"ISO8859-3" "ISO8859-4" "ISO8859-5" "TIS11" "TIS18"
 *
 */
- (void)cpc1setCharacterCodePage:(NSString *_Nonnull)codepage;
```

- 参数

参数	类型	描述	范围
codepage	NSString	对应的代码页字符串	查看上面注释

### 3.39 阿拉伯语变形

- 描述

- 设置阿拉伯语变形，需要重启打印机
- 查询阿拉伯语变形方式

- 代码

```

/*!
 *  \~chinese
 *
 *  阿拉伯语变形,设置后重启打印机
 *
 *  @param function 48: 关闭; 49: 变形1, 按单词排序; 50: 变形2, 按短语排序; 51: 变形
3, 按完整规则排序
 *
 *  \~english
 *
 *  Arabic transform
 *
 *  @param function 48: close; 49: Sort by word; 50: Phrase sort; 51: Sort by
complete rules
 *
 */
- (void)cpc1SetArabicTransformFunction:(NSInteger)function;

/*!
 *  \~chinese
 *
 *  查询阿拉伯语变形状态, 00 00: 关闭; 01 00: 变形1; 02 00: 变形2; 03 00: 变形3
 *
 *  \~english
 *
 *  get Arabic status:00 00: close; 01 00: sort by word; 02 00: Phrase
sort; 03 00: sort by complete rules
 *
 */
- (void)cpc1GetArabicTransformStatus;

```

- 参数

- 设置变形方式

参数	类型	描述	范围
function	NSInteger	变形方式	48: 关闭 49: 按单词排序 50: 按短语排序 51: 按完整规则排序

- 返回值

- 获取变形方式，返回的是十六进制数据,通过在 `PTDispatcher` 的接口可接收返回的数据

```

/*!
 * \~chinese
 * 接收到数据回调
 *
 * @param receiveDataBlock 回调block
 *
 * \~english
 * Received data callback
 *
 * @param receiveDataBlock Trigger block
 */
- (void)whenReceiveData:(PTDataParameterBlock)receiveDataBlock;

```

返回值，十六进制数据	描述
00 00	关闭
01 00	按单词排序
02 00	按短语排序
03 00	按完整规则排序

### 3.40 泰文变形

- 描述

设置泰文变形是否开启

- 代码

```
- (void)cpc1setThaiTransformStatus:(NSInteger)status;
```

- 参数

参数	类型	描述	范围
status	NSInteger	是否开启泰文变形	48: 关闭 49:开启

### 3.41 越南语变形

- 描述

- 设置越南语变形方式
- 获取越南语变形方式

- 代码

```
/*!
 * \~chinese
 *
 * 越南语变形, 设置后重启打印机.
 *
 * @param function 48: 关闭; 49: ASCII输入; 50: UTF-8输入
 *
 * \~english
 *
 * Set Vietnamese Transform
 *
 * @param function 48: close; 49: ASCII; 50: UTF-8
 *
 */
- (void)cpclSetVietnameseTransformFunction:(NSInteger)function;

/*!
 * \~chinese
 *
 * 查询越南语变形状态, 00 00: 关闭; 01 00: ASCII输入; 02 00: UTF-8输入;
 *
 * \~english
 *
 * get Vietnamese status:00 00: close; 01 00: ASCII; 02 00: UTF-8;
 *
 */
- (void)cpclGetVietnameseTransformStatus;
```

- 参数

- 设置越南语变形

参数	类型	描述	范围
function	NSInteger	越南语变形的类型	48: 关闭 49: ASCII输入 50: UTF-8输入

- 返回值

- 获取越南语变形方式, 返回的是十六进制数据, 通过在 `PTDispatcher` 的接口可接收返回的数据

```

/**!
 *  \~chinese
 *  接收到数据回调
 *
 *  @param receiveDataBlock 回调block
 *
 *  \~english
 *  Received data callback
 *
 *  @param receiveDataBlock  Trigger block
 */
- (void)whenReceiveData:(PTDataParameterBlock)receiveDataBlock;

```

返回值, 十六进制数据	描述
00 00	关闭
01 00	ASCII输入
02 00	UTF-8输入

### 3.42 获取固件版本

- 描述

- 获取打印机的固件版本号
- 并非所有机型都支持

- 代码

```
- (void)getFirmwareVersion;
```

- 返回值

- 返回的是十六进制数据, 通过在 `PTDispatcher` 的接口可接收返回的数据

```

/*!
 *  \~chinese
 *  接收到数据回调
 *
 *  @param receiveDataBlock 回调block
 *
 *  \~english
 *  Received data callback
 *
 *  @param receiveDataBlock Trigger block
 */
- (void)whenReceiveData:(PTDataParameterBlock)receiveDataBlock;

```

- 返回的是一个以5f开头，00结尾的十六进制数据
- 去掉头尾后，可用 `+ (NSString *)decodeStringWithData:(NSData *)data;` 解析

```

/*!
 *  \~chinese
 *  解码,default:GBK
 *
 *  @param data          需要解码的数据
 *
 *  @return 解码后的字符串
 *
 *  \~english
 *  Decoding,default:GBK
 *
 *  @param data          Data to be decoded
 *
 *  @return Decoded data
 */
+ (NSString *)decodeStringWithData:(NSData *)data;

```