

目录

目录.....	1
简介.....	3
集成说明.....	4
接口说明.....	5
CAPrintCommon.....	6
getPrinterStatus.....	7
getPrinterResolution.....	8
printSelfTestPage.....	9
printImage.....	10
writeBuffer.....	12
queryPrinterBatteryLevel.....	13
queryPrinterSerialNumber.....	14
CA_Proto_SetSystemNameAndSerialNumber.....	15
CA_Proto_SetBluetoothNameAndBluetoothPassword.....	16
CA_Proto_SetPTPBasicParameters.....	17
CA_Settings_Hardware_SetPrintSpeed.....	20
CA_Settings_DownloadFirmware.....	21
CAPrintResult.....	22
CAPrinterConnector.....	23
connectPrinterAsync.....	24
connectPrinterSync.....	25
disconnectPrinter.....	26
getCurrentPrinterDevice.....	27
getCurrentPrinterHandle.....	28
getCurrentConnectionStatus.....	29
isCurrentConnectingPrinter.....	30
isCurrentConnectedPrinter.....	31
isCurrentDisconnectingPrinter.....	32
isCurrentDisconnectedPrinter.....	33
isCurrentConnectionValid.....	34
registerConnectionStatusChangedEvent.....	35
unregisterConnectionStatusChangedEvent.....	36
CAPrinterDevice.....	37
CAPrinterDiscover.....	38
startDiscover.....	39
stopDiscover.....	40
isDiscovering.....	41
setOnPrinterDiscoveredListener.....	42
setOnPrinterDiscoverThreadListener.....	43
CAPrinterResolution.....	44
CAPrinterStatus.....	45

简介

此开发包为了简化打印，便于使用者集成打印功能，精简并去掉了很多不常用的功能，仅保留了基础功能。主要有：搜索连接打印机，获取打印机状态，打印图片，可选择打印完是否开钱箱，是否进纸，是否切刀，是否等待打印完成。

集成说明

1 引入 `caprint.framework` 即可，不会引用的可以去网上搜索一下

接口说明

本开发包的全部内容，均在 caprint 包里，主要功能：

1 搜索打印机

使用 CAPrinterDiscover 搜索打印机，搜到的打印机用 CAPrinterDevice 表示。

2 连接打印机

使用 CAPrinterConnector 连接打印机，连接时传入参数 CAPrinterDevice 即可。

3 打印图片

使用 CAPrintCommon，打印时传入参数 CAPrinterConnector 和要打印的图片等参数即可。

CAPrintCommon

通用打印类，所有机型均可调用该类中的函数。

主要有获取打印机状态，获取打印机分辨率，打印自检页，打印图片等功能。

getPrinterStatus

获取打印机状态，如果打印机未连接，获取失败，则返回 null。

定义

```
public static func getPrinterStatus(printerConnector: CAPrinterConnector) -> CAPrinterStatus?
```

getPrinterResolution

获取打印机分辨率，如果打印机未连接，获取失败，则返回 null。

定义

```
public static func getPrinterResolution(printerConnector: CAPrinterConnector) -> CAPrinterResolution?
```


printSelfTestPage

打印自检页，自检页上会显示打印机软件版本，还有其他有用的信息，便于出现问题时，快速排查问题。

定义

```
public static func printSelfTestPage(printerConnector: CAPrinterConnector) -> Bool
```

printImage

打印图片，并指定打印图片前后的相关动作

定义

```
public static func printImage(printerConnector: CAPrinterConnector, image: UIImage, binaryzationMethod: Int, compressionMethod: Int, paperType: Int, printAlignment: Int, printSpeed: Int, printDensity: Int, kickDrawer: Bool, feedPaper: Double, cutPaper: Int, waitPrintFinished: Int) -> CAPrintResult
```

参数

printerConnector

一个打印机连接

image

要打印的图片，注意图片宽度不要超过了打印机可打印区域，超了可能会乱码

binaryzationMethod

图片二值化算法

0 抖动算法

1 阈值算法

2 误差扩散法

compressionMethod

图片压缩算法

0 不压缩

1 压缩算法 1

2 压缩算法 2

paperType

指定纸张类型，如果指定了标签纸或黑标纸，则打印完会自动定位到缝隙处

1 连续纸

2 标签纸

3 黑标纸

printAlignment

指定对齐方式，如果图片宽度小于可打印宽度，不指定的话，可能打偏

0 左对齐

1 居中对齐

2 右对齐

printSpeed

指定打印速度，0 表示不指定打印速度，大于零表示指定打印速度为多少 mm/s。

（蓝牙打印速度会根据数据内容自动调整）

printDensity

指定打印浓度，浓度范围[0-15]。如果设置为-1，则不指定打印浓度。

kickDrawer

打印前是否开钱箱，由于打印需要时间，如果打印后开钱箱，会影响效率，一般都是打印前开钱箱

feedPaper

打印完图片，额外进纸毫米，主要是多走一点纸，便于撕纸

cutPaper

打印完图片，进完纸之后，进行切刀，有切刀的机器，用切刀切纸，不需要专门去撕纸。

0 是不切纸，1 是全切，2 是半切。

waitPrintFinished

是否等待打印完成。

取值 0 表示数据发送完成直接返回，不等待打印完成；

取值大于 0 表示，最多等待对应的毫秒时间直到打印完成，

如果在超时时间内打印完成，则返回 true，如果超时到了还没打完，返回 false。

返回

数据写入成功，返回 true，数据写入失败，返回 false。

如果设置了等待打印完成，则超时时间内打印完成，才返回 true，否则返回 false。

其他字段还记录了一些数据发送情况，请看 CAPrintResult 对应的字段说明。

writeBuffer

写数据到打印机

定义

```
public static func writeBuffer(printerConnector: CAPrinterConnector, buffer: [UInt8]) -> Int
```

参数

printerConnector

一个打印机连接

buffer

要写入的数据

返回

返回写入的字节数，-1 表示写入失败。

queryPrinterBatteryLevel

查询打印机电量，返回值为电量百分比，返回-1 表示查询失败。

定义

```
public static func queryPrinterBatteryLevel(printerConnector: CAPIPrinterConnector) -> Int
```

queryPrinterSerialNumber

查询打印机序列号

定义

```
public static func queryPrinterSerialNumber(printerConnector: CAPIPrinterConnector) -> String?
```

CA_Proto_SetSystemNameAndSerialNumber

设置打印机系统名称和序列号，设置完需要重启打印机生效。

定义

```
public static func CA_Proto_SetSystemNameAndSerialNumber(printerConnector: CAPrinterConnector, systemName: String,
serialNumber: String) -> Bool
```

CA_Proto_SetBluetoothNameAndBluetoothPassword

设置打印机蓝牙名称和蓝牙密码，设置完需要重启打印机生效。

定义

```
public static func CA_Proto_SetBluetoothNameAndPassword(printerConnector: CAPrinterConnector, bluetoothName: String, bluetoothPassword: String) -> Bool
```


CA_Proto_SetPTPBasicParameters

设置基础信息，包括语言、波特率、浓度等参数，对应设置工具 PTP 页。

定义

```
public static func CA_Proto_SetPTPBasicParameters(printerConnector: CAPrinterConnector, baudrate: Int, codepage: Int,
density: Int, asciiFontType: Int, lineFeed: Int, idleTime: Int, powerOffTime: Int, maxFeedLength: Int, pageLength: Int)
-> Bool
```

参数

//// 设置基础信息，包括语言、波特率、浓度等参数，对应设置工具 PTP 页。

////

//// printerConnector

//// 打印机连接

////

//// baudrate

//// 要设置的波特率

////

//// codepage

//// 要设置的语言

//// 具体语言对应的数值如下：

//// { ("简体中文"), 255 },

//// { ("繁體中文"), 254 },

//// { ("UTF - 8"), 253 },

//// { ("SHIFT - JIS"), 252 },

//// { ("EUC - KR"), 251 },

//// { ("CP437[U.S.A., Standard Europe]"), 0 },

//// { ("Katakana"), 1 },

//// { ("CP850[Multilingual]"), 2 },

//// { ("CP860[Portuguese]"), 3 },

//// { ("CP863[Canadian - French]"), 4 },

//// { ("CP865[Nordic]"), 5 },

//// { ("WCP1251[Cyrillic]"), 6 },

//// { ("CP866 Cyrillic #2"), 7 },

//// { ("MIK[Cyrillic / Bulgarian]"), 8 },

//// { ("CP755[East Europe, Latvian 2]"), 9 },

//// { ("Iran"), 10 },

//// { ("CP862[Hebrew]"), 15 },

//// { ("WCP1252 Latin I"), 16 },

//// { ("WCP1253[Greek]"), 17 },

//// { ("CP852[Latina 2]"), 18 },

//// { ("CP858 Multilingual Latin I + Euro"), 19 },

//// { ("Iran II"), 20 },

//// { ("Latvian"), 21 },

//// { ("CP864[Arabic]"), 22 },

```

/// { ("ISO - 8859 - 1[West Europe]", 23 },
/// { ("CP737[Greek]", 24 },
/// { ("WCP1257[Baltic]", 25 },
/// { ("Thai", 26 },
/// { ("CP720[Arabic]", 27 },
/// { ("CP855", 28 },
/// { ("CP857[Turkish]", 29 },
/// { ("WCP1250[Central Eurpoe]", 30 },
/// { ("CP775", 31 },
/// { ("WCP1254[Turkish]", 32 },
/// { ("WCP1255[Hebrew]", 33 },
/// { ("WCP1256[Arabic]", 34 },
/// { ("WCP1258[Vietnam]", 35 },
/// { ("ISO - 8859 - 2[Latin 2]", 36 },
/// { ("ISO - 8859 - 3[Latin 3]", 37 },
/// { ("ISO - 8859 - 4[Baltic]", 38 },
/// { ("ISO - 8859 - 5[Cyrillic]", 39 },
/// { ("ISO - 8859 - 6[Arabic]", 40 },
/// { ("ISO - 8859 - 7[Greek]", 41 },
/// { ("ISO - 8859 - 8[Hebrew]", 42 },
/// { ("ISO - 8859 - 9[Turkish]", 43 },
/// { ("ISO - 8859 - 15[Latin 3]", 44 },
/// { ("Thai2", 45 },
/// { ("CP856", 46 },
/// { ("Cp874", 47 },
/// { ("Other(Vietnam)", 48 },
///
/// density
/// 要设置的打印浓度
/// 0 - Light
/// 1 - Normal
/// 2 - Dark
///
/// asciiFontType
/// 要设置的英文字体类型
/// 0 - FontA(12x24)
/// 1 - FontB(9x24)
/// 2 - FontC(9x17)
/// 3 - FontD(8x16)
///
/// lineFeed
/// 换行
/// 0 - LF(0x0A)
/// 1 - CR(0x0D)
///
/// idleTime
/// 空闲时间 (秒数)
///
/// powerOffTime

```

/// 关机时间 (秒数)

///

```
/// maxFeedLength
```

/// 最大进纸距离 (毫米)

///

/// pageLength

/// 页面长度 (毫米)

/// /// ///

```
/// return
```

/// 返回值仅指示指令是否写入成功。返回 true 表示写入成功，返回 false 表示写入失败。

CA_Settings_Hardware_SetPrintSpeed

设置打印速度

定义

```
public static func CA_Settings_Hardware_SetPrintSpeed(printerConnector: CAPIPrinterConnector, nSpeed: Int) -> Bool
```

参数

```
/// 设置打印速度
```

```
///
```

```
/// nSpeed
```

```
/// 打印速度，单位毫米每秒
```

CA_Settings_DownloadFirmware

下载固件

定义

```
public typealias ca_download_progress_block_t = (_ total_length: Int, _ writed_length: Int) -> Void
public static func CA_Settings_DownloadFirmware(printerConnector: CAPrinterConnector, firmware: [UInt8], progressBlock:
ca_download_progress_block_t?) -> Bool
```

参数

```
//// 下载固件
////
//// firmware
////     完整固件，不可多不可少。必须匹配对应的机器。
//// progressBlock
////     显示下载进度
////     total_length
////     命令的总长度（比固件多了 6 个字节的头）
////     writed_length
////     已发送的命令长度
```

CAPrintResult

用于描述打印结果，并记录相关打印数据信息。

返回的结果包括，是否打印成功，蓝牙发包情况，数据发送总量，数据发送耗时，数据发送速率。

```
public class CAPrintResult {  
  
    /// 打印结果，是成功还是失败  
    public var printResult = false  
  
    /// 打印的图片高度，是多少毫米  
    public var sourceImageHeight: Double = 0  
  
    /// 设置的打印速度，是多少毫米每秒  
    public var settedPrintSpeed = 0  
  
    /// 本次打印发送的数据量，是多少个字节数  
    public var writeDataSize = 0  
  
    /// 本次打印发送数据耗时，毫秒时间  
    public var writeTimeUsed: Double = 0  
  
    /// 本次打印发送数据的速率，kb/s  
    public var writeDataSpeed: Double = 0  
  
    public init()  
  
    public func toString() -> String  
  
}
```

CAPrinterConnector

打印机连接类，用来连接和断开一个打印机设备。

如果注册连接状态回调，则可以在连接状态变化时，获得通知。

若设备端关闭，此处也可以及时知道，并触发连接状态变化回调。

蓝牙打印机关闭，大约 2s 可以知道。

网络打印机关闭，大约 30s 可以知道。

connectPrinterAsync

启动一个子线程去连接打印机设备（异步），连接结果需要通过 [连接状态变化回调接口](#) 获取。
返回值仅表示本次是否启动子线程。

定义

```
public func connectPrinterAsync(printerDevice: CAPrinterDevice) -> Bool
```


connectPrinterSync

直接连接打印机设备（同步），返回连接结果。

定义

```
public func connectPrinterSync(printerDevice: CAPrinterDevice) -> Bool
```

disconnectPrinter

断开打印机连接

定义

```
public func disconnectPrinter()
```

getCurrentPrinterDevice

获取当前打印机设备

定义

```
public func getCurrentPrinterDevice() -> PrinterDevice?
```

getCurrentPrinterHandle

获取当前打印机连接句柄

定义

```
public func getCurrentPrinterHandle() -> Int
```

getCurrentConnectionStatus

获取当前连接状态

定义

```
public func getCurrentConnectionStatus() -> ConnectionStatus
```

isCurrentConnectingPrinter

是否正在连接打印机

定义

```
public func isCurrentConnectingPrinter() -> Bool
```

isCurrentConnectedPrinter

是否已连接打印机

定义

```
public func isCurrentConnectedPrinter() -> Bool
```

isCurrentDisconnectingPrinter

是否正在断开打印机

定义

```
public func isCurrentDisconnectingPrinter() -> Bool
```


isCurrentDisconnectedPrinter

是否已断开打印机

定义

```
public func isCurrentDisconnectedPrinter() -> Bool
```

isCurrentConnectionValid

连接是否有效。若长时间(6-8 秒)未收到打印机自动回复的状态，即认为链接失效了。需要重新连接打印机。

定义

```
public func isCurrentConnectionValid() -> Bool
```

registerConnectionStatusChangedEvent

注册连接状态变化回调

定义

```
public func registerConnectionStatusChangedEvent(callback: @escaping OnConnectionStatusChanged, tag: Int) -> Bool
```

unregisterConnectionStatusChangedEvent

移除连接状态变化回调

定义

```
public func unregisterConnectionStatusChangedEvent(tag: Int) -> Bool
```

CAPrinterDevice

描述打印机设备。
有打印机名称，端口类型，端口地址等属性。

CAPrinterDiscover

打印机搜索类，用于搜索打印机设备。
目前能搜索到蓝牙打印机，网络打印机。

startDiscover

开始搜索打印机，搜到的打印机会通过回调接口告知。

定义

```
public func startDiscover()
```

stopDiscover

停止搜索打印机

定义

```
public func stopDiscover()
```


isDiscovering

是否正在搜索打印机

定义

```
public func isDiscovering() -> Bool
```

setOnPrinterDiscoveredListener

设置搜到打印机回调接口

定义

```
public func setOnPrinterDiscoveredListener(onPrinterDiscoveredListener: @escaping OnPrinterDiscovered)
```

setOnPrinterDiscoverThreadListener

设置搜索线程开始和结束回调

定义

```
public          func          setOnPrinterDiscoverThreadListener(onPrinterDiscoverThreadStartedListener:          @escaping  
OnPrinterDiscoverThreadStarted, onPrinterDiscoverThreadFinishedListener: @escaping OnPrinterDiscoverThreadFinished)
```

CAPrinterResolution

描述打印机可打印区域的宽度，每毫米打印点数。
对于标签打印机来说，还可得到最大支持的标签页面高度。

CAPrinterStatus

描述打印机状态。

包含错误状态，信息状态。

错误状态，表明打印机出错，需要排除错误后打印机才可以继续打印。

信息状态，仅提供一些打印机的信息，比如是否处于标签模式等。

```
// ERROR_CUTTER
//      切刀错误
// ERROR_FLASH
//      串行 FLASH 错误
// ERROR_NOPAPER
//      缺纸
// ERROR_VOLTAGE
//      电压错误
// ERROR_MARKER
//      黑标或缝标侦测错误
// ERROR_ENGINE
//      机芯未识别
// ERROR_OVERHEAT
//      过热
// ERROR_COVERUP
//      开盖或轴未压下
// ERROR_MOTOR
//      马达失步(一般为卡纸)
// INFO_LABELPAPER
//      当前纸张识别为标签纸(为 0 是连续纸)
// INFO_LABELMODE
//      当前为标签模式
// INFO_HAVEDATA
//      有数据开始处理
// INFO_NOPAPERCANCELED
//      上次单据是缺纸后取消了
// INFO_PAPERNOFETCH
//      单据未取走
// INFO_PRINTIDLE
//      当前打印空闲
// INFO_RECVIDLE
//      当前接收缓冲区为空
```

