

Contents

Contents.....	1
Introduction.....	3
Integrated description.....	4
Interface Specification.....	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

Introduction

In order to simplify printing and facilitate users to integrate printing functions, this development kit simplifies and eliminates many uncommon functions, and only retains the basic functions, including searching and connecting the printer, obtaining the status of the printer, printing image, and choosing whether to open the cash box, enter the paper, cut the knife and wait for the completion of printing.

Integrated description

1 simply reference `caprint.framework`, if you don 't know, google it.

Interface Specification

all sdk contents is contained in package caprint. supports:

1 Search Printer

Use CAPrinterDiscover to search printer, The found printer is represented by CAPrinterDevice.

2 Connect Printer

Use CAPrinterConnector to connect printer, Pass in parameters CAPrinterDevice when connecting.

3 Print Image

Use CAPrintCommon, Pass in parameters CAPrinterConnector & image when printing.

CAPrintCommon

General purpose print class, all models can call the functions in this class

It mainly has the functions of obtaining the printer state, obtaining the printer resolution, printing the self-check page and printing the image.

getPrinterStatus

Gets the status of the printer and returns NULL if the printer is not connected

Syntax

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

getPrinterResolution

Gets the printer resolution and returns NULL if the printer is not connected

Syntax

```
public static func getPrinterResolution(printerConnector: CAPIPrinterConnector) -> CAPIPrinterResolution?
```


printSelfTestPage

Print the self-check page, which displays the printer software version and other useful information for quick troubleshooting when a problem occurs

Syntax

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

printImage

Print the image and specify the actions before and after printing the image

Syntax

```
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
```

Parameters

printerConnector

A printer connection

image

To print the image, pay attention to the image width does not exceed the printer printable area, too much may be messy

binaryzationMethod

Image binaryzation method

0 dithering

1 thresholding

2 error diffusion

compressionMethod

Image compression method

0 don't compress

1 compression method 1

2 compression method 2

paperType

Specifies the paper type. If label paper or black label paper is specified, it will automatically position to the gap after printing

1 Serial Paper

2 Label Paper

3 Black Marker Paper

printAlignment

Specify the alignment. If the picture width is less than the printable width, it may be biased if it is not specified

0 align left

1 align center

2 align right

printSpeed

Specify the printing speed, 0 means no printing speed is specified, and greater than zero means the specified printing speed is mm / s.

(the Bluetooth printing speed will be automatically adjusted according to the data content)

printDensity

Specify print concentration, concentration range [0-15]. If set to - 1, the print density is not specified.

kickDrawer

Whether to open the money box before printing? Because printing takes time, if the money box is opened after printing, the efficiency will be affected. Generally, it is opened before printing

feedPaper

After printing the picture, feed extra paper in millimeters, mainly take more paper, easy to tear the paper

cutPaper

After printing pictures, into the paper, the knife, a knife machine, with the knife to cut paper, there is no need to tear paper

0 means no cut, 1 means full cut, 2 means half cut.

waitPrintFinished

Whether to wait for printing to complete

The value 0 indicates that the data is sent and returned directly without waiting for the printing to complete

A value greater than 0 indicates a maximum wait of milliseconds until the printing is complete

This returns true if the print completes within the timeout period and false if the timeout has not yet been completed

Return value

Data write success returns true, data write failure returns false

If you set the wait for printing to complete, then the timeout period for printing to complete will return true, otherwise it returns false

Other fields record data send information, please see CAPrintResult for detail

writeBuffer

Write data to printer

Syntax

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

Parameters

printerConnector

A printer connection

buffer

buffer

Return value

return bytes writted. or return -1 means failed

queryPrinterBatteryLevel

Query printer battery level, return percent value, return -1 means query failed.

Syntax

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

queryPrinterSerialNumber

Query printer serial number

Syntax

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

CA_Proto_SetSystemNameAndSerialNumber

Set printer system name and serial number. Need reboot printer to effect.

Syntax

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

CA_Proto_SetBluetoothNameAndBluetoothPassword

Set printer bluetooth name and bluetooth password. Need reboot printer to effect.

Syntax

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


CA_Proto_SetPTPBasicParameters

Set basic parameters, include codepage,baudrate,density, like printersetting.exe ptp page.

Syntax

```
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
```

Parameters

```
//      Set basic parameters, include codepage,baudrate,density, like printersetting.exe ptp page.
//
// printerConnector
//      printer connector
//
// baudrate
//      the baudrate to set
//
// codepage
//      the codepage to set
//      see following:
//      { ("Simplified Chinese"), 255 },
//      { ("Traditional Chinese"), 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 Cyrilliec #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
//      the density to set
//      0 - Light
//      1 - Normal
//      2 - Dark
//
// asciiFontType
//      the ascii text font type
//      0 - FontA(12x24)
//      1 - FontB(9x24)
//      2 - FontC(9x17)
//      3 - FontD(8x16)
//
// lineFeed
//      the line feed char
//      0 - LF(0x0A)
//      1 - CR(0x0D)
//
// idleTime
//      idle time (seconds)
//
// powerOffTime

```

```
//      power off time (seconds)
//
// maxFeedLength
//      max feed length (mm)
//
// pageLength
//      page length (mm)
//
// return
//      If command is written successfully, it returns true else it returns false.
```

CA_Settings_Hardware_SetPrintSpeed

set print speed

Syntax

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

Parameters

```
//      set print speed
//
// nSpeed
//      print speed in mm/s
//
// return
//      If command is written successfully, it returns true else it returns false.
```

CA_Settings_DownloadFirmware

Download firmware

Syntax

```
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
```

Parameters

```
/// Download firmware
///
/// firmware
///     the firmware. Must match the machine.
/// progressBlock
///     show download progress
///     total_length
///         cmd total length
///     writed_length
///         cmd sended length
```

CAPrintResult

Used to describe the printing results and record relevant printing data information.

The returned results include whether the printing is successful, the bluetooth packet situation, the total amount of data sent, the time of data sent, and the data sent rate.

This class was generated when the Boss asked to record the Bluetooth packet and see if there was a resend packet.

```
public class CaPrintResult {

    /// the print result, success or failure
    public var printResult = false

    /// The height of the printed picture, how many millimeters
    public var sourceImageHeight: Double = 0

    /// Set the printing speed to millimeters per second
    public var settedPrintSpeed = 0

    /// How many bytes of data will be sent this time
    public var writeDataSize = 0

    /// It takes milliseconds to print and send data
    public var writeTimeUsed: Double = 0

    /// The rate at which data is sent for this printing, kb/s
    public var writeDataSpeed: Double = 0

    public init()

    public func toString() -> String

}
```

CAPrinterConnector

Printer connection class used to connect and disconnect a printer device

If you register a connection status callback, you can be notified when the connection status changes

If the device is closed, this is also a good place to know and trigger a connection status change callback

Bluetooth printer off, about 2S can know

TCP printer off, about 30s can know

connectPrinterAsync

Start a child thread to connect to the printer device (asynchronously). The result of the connection needs to get the return value through the connection status change callback interface, which only indicates whether the child thread is started this time

Syntax

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


connectPrinterSync

Connect directly to the printer device (sync) and return the connection result

Syntax

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

disconnectPrinter

disconnect printer connection

Syntax

```
public func disconnectPrinter()
```

getCurrentPrinterDevice

get current printer device

Syntax

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

getCurrentPrinterHandle

get current printer handle

Syntax

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

getCurrentConnectionStatus

get current connection status

Syntax

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

isCurrentConnectingPrinter

is current connecting printer

Syntax

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

isCurrentConnectedPrinter

is current connected printer

Syntax

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

isCurrentDisconnectingPrinter

is current disconnecting printer

Syntax

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


isCurrentDisconnectedPrinter

is current disconnected printer

Syntax

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

isCurrentConnectionValid

Whether the connection is valid If you do not receive an automatic reply from the printer for a long time (6–8 seconds), the link is invalid and you need to reconnect the printer

Syntax

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

registerConnectionStatusChangedEvent

register connection status changed event

Syntax

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

unregisterConnectionStatusChangedEvent

remove connection status changed event

Syntax

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

CAPrinterDevice

Description printer equipment

There are printer name, port type, port address and other properties

CAPrinterDiscover

Printer search class for searching printer devices

Bluetooth printer ,TCP network printer can be searched

startDiscover

Start searching for the printer, and the printer will be notified via the callback interface

Syntax

```
public func startDiscover()
```

stopDiscover

stop searching printer

Syntax

```
public func stopDiscover()
```


isDiscovering

is searching printer

Syntax

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

setOnPrinterDiscoveredListener

set printer discovered listener

Syntax

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

setOnPrinterDiscoverThreadListener

set discover thread listener

Syntax

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

CAPrinterResolution

Describes the width of the printer's printable area, with print points per millimeter
For tag printers, you can also get the maximum supported tag page height

CAPrinterStatus

Description of printer status

Contains error status, information status

The error status indicates that the printer is in error, and the printer cannot continue printing until the error is removed

Information state, which provides only some information about the printer, such as whether it is in tag mode, etc

```
// ERROR_CUTTER
//      Cutter error
// ERROR_FLASH
//      FLASH error
// ERROR_NOPAPER
//      No paper
// ERROR_VOLTAGE
//      Voltage error
// ERROR_MARKER
//      Black mark or seam mark error detected
// ERROR_ENGINE
//      Unrecognized printer engine
// ERROR_OVERHEAT
//      Overheat
// ERROR_COVERUP
//      Open cover or shaft not pressed down
// ERROR_MOTOR
//      Motor out of step (usually paper jam)
// INFO_LABELPAPER
//      Current paper identified as label paper (0 is continuous paper)
// INFO_LABELMODE
//      Currently in label mode
// INFO_HAVEDATA
//      We have data to start processing
// INFO_NOPAPERCANCELED
//      The last document was cancelled after it was short of paper
// INFO_PAPERNOFETCH
//      The documents were not taken
// INFO_PRINTIDLE
//      Current print idle
// INFO_RECVIDLE
//      The current receive buffer is empty
```

