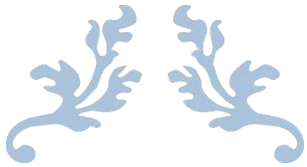

IOSSDK INSTRUCTIONS

Contents

Chapter 1 SDK Instructions

1.1 Introduction to IOSSDK

To use the SDK, you need to add system dependency libraries: SystemConfiguration.framework, CFNetwork.framework, CoreBluetooth.framework.

In the sdk file of IosSdkSourceFile, the visible header files (.h files) are: PosBLEManager.h, PosWIFIManager.h, TscCommand.h, PosCommand.h, ImageTranster.h and POSSDK.h. These .h files include the connection of Bluetooth and wifi: BLEMananger.h and WIFIManager, the methods of sending and receiving data and the agents that need to be followed, as well as the tool classes for instruction encapsulation: TscCommand, PosCommand, and image processing class: ImageTranster.

PosBLEManager.h is the Bluetooth management class, which handles Bluetooth connection related and POS command sending.

Use the [PosBLEManager sharedInstance] singleton method to create a management object, follow the proxy while creating it, and implement the proxy method. Call the startScan method to start scanning, and get the scan result in the proxy method didUpdatePeripheralList. connectDevice: is the Bluetooth connection method to connect to the specified peripheral. There is a writePeripheral property in PosBLEManager, which is used to specify which peripheral to write data to, and does not specify the peripheral that is connected to the default bit last.

PosWIFIManager.h is the wifi management class, which handles the wifi connection and the sending of barcode commands. A single connection can use the singleton method [PosWIFIManager shareWifiManager] to create a connection object and follow the proxy, ConnectWithHost:port:completion: is the connection method, specify the IP and port number, and whether the block callback is successful. For

multiple connections, use the [[PosWIFIManager alloc] init] method to initialize multiple management objects, save them, and use the corresponding objects to send instructions.

TscCommand.h barcode command packaging tool class, all return values are of NSData type; PosCommand.h, pos command packaging tool classes, return are all NSData type. The methods in these two instruction tool classes are class methods, which are called directly by the class name, and the return value is of NSData type, which can be used to send data directly. The realization of Bluetooth and WiFi connection needs to follow the proxy, please refer to the sample code for details.

The specific method of sending data, in PosBLEManager.h and PosWIFIManager.h, is -(void)WriteCommandWithData:(NSData *)data; and the method with callback -(void)WriteCommandWithData:(NSData*)data
callBack:(TSCCompletionBlock)block.

1.2 Instructions for Bluetooth Management

First of all: use the [PosBLEManager sharedInstance] singleton method to create a management object, set a proxy, and implement the proxy method. Note: When the PosBLEManager object is created, Bluetooth has been turned on to scan, and the scan results have also been saved. At this point, in the implemented proxy method-(void)PosdidUpdatePeripheralList:(NSArray *)peripherals RSSIList:(NSArray *)rssiList;, you can get the scan results, and you can connect through the scanned device, PosconnectDevice:; the connection is successful Or failure will call the proxy method-(void)PosdidConnectPeripheral:(CBPeripheral *)peripheral; and-(void)PosdidFailToConnectPeripheral:(CBPeripheral *)peripheral error:(NSError *)error; method.

After the connection is successful, you can call the method of sending data after

obtaining the characteristics of writing data. It is recommended to use the PosWriteCommandWithData: method to send. After the data is sent, the proxy method PosdidWriteValueForCharacteristic:(CBCharacteristic *)character error:(NSError *)error will be called back; error can be used to determine whether the data is sent successfully, and then do the corresponding operations. You can also use the PosWriteCommandWithData:callblock: method to send, and use block to call back whether it is successful.

1.3 Wi-Fi management instructions

A single connection can use the singleton method [PosWIFIManager shareWifiManager] to create a connection object, set a proxy, and implement the proxy method.
PosConnectWithHost:port:completion: is the connection method, specify the IP and port number, and whether the block callback is successful. The proxy method can also be implemented-(void)PosWIFIManager:(PosWIFIManager *)manager didConnectedToHost:(NSString *)host port:(UInt16)port; to handle operations after connection success or failure.
To
After the connection is successful, it is recommended to use
-(void)PosWriteCommandWithData:(NSData *)data; and
To
-(Void)PosWriteCommandWithData:(NSData *)data
withResponse:(PosWIFICallBackBlock)block;
To send data, you can use block callbacks for the result of sending data, or perform the operation after judgment in the proxy method of PosWIFIManager:(PosWIFIManager *)manager didWriteDataWithTag:(long)tag.