Description Document for Guidance on Printer SDK Connection

1、usage

Put in the target project the SDK documents which comprise libXYPrinter.a and visible header files (.h). Visible header files are divided as the diagram below:

Note： Please conduct the test in the real machine environment since the analog simulator cannot be tested. In case of any error, please import the following system framework:

SystemConfiguration.framework
CFNetwork.framework
CoreBluetooth.framework

2、header description

BLEManager.h	Bluetooth connection
WIFIManager.h	WiFi connection
TscCommand.h	Tsc command class
PosCommand.h	Thermal printer command class
ImageTranster.h	Image processing class

2.1 Description of Bluetooth connection

BLEManager.h is of Bluetooth management type which deals with Bluetooth connection and sends POS commands.

You may use [BLEManager sharedInstance] singletion method to create the management object while following the proxy to implement the proxy method. You may call up startScan method to begin the scanning and obtain the scanning result with the proxy method didUpdatePeripheralList.

connectDevice is a Bluetooth connection to connect the designated peripheral devices. BLEManager is provided with an attribute of writePeripheral to designate which peripheral device data is written to. If no device is designated, the peripheral device last connected is the default one.

2.2 Description of WiFi connection

WIFIManager.h is of wifi management type which deals with wifi connection and sending of barcode command. For single connection, the singletion method [WIFIManager shareWifiManager] is used to create the connection object and follow the proxy. ConnectWithHost: port: completion: is the connection method. Designate IP and port No. and determine whether it is successful in case of block callback.

In case of multiple connection, the [[WIFIManager alloc] init] method may initialize multiple management objects and be saved. Corresponding objects are used to send commands.

2.3 Description of command

TscCommand.h ：
For TscCommand.h barcode command package tool, all return values are of NSData type;
PosCommand.h:
For PosCommand.h,pos command package tool, all return values of NSData type.

Methods in the above two command tools are of class method which may be called up by class and all return values are of NSData type that may be used for direct data sending.

Implementation of Bluetooth and WiFi connection needs to follow proxy. For details, refer to the code example.

3. Usage of Bluetooth management

Firstly, use the [BLEManager sharedInstance] singletion method to create management objects, set proxy and implement the proxy method. Note: once the object of BLEManager is created, Bluetooth may have enable the scanning with scanning result being saved. At the time, in the proxy method as implemented - (void)didUpdatePeripheralList:(NSArray *)peripherals RSSIList:(NSArray *)rssiList,it is possible to get the scanning result and establish the connection through the scanning device, connectDevice:; connection success or failure may call up the proxy

method
-(void)didConnectPeripheral:(CBPeripheral*)peripheral;and
- (void)didFailToConnectPeripheral:(CBPeripheral *)peripheral error:(NSError *)error; method.

Once the connection is successful, it is possible to call up the method of data sending to obtain the characteristics of the data as written. WriteCommandWithData: is recommended for sending. Once data is sent, the proxy method didWriteValueForCharacteristic:(CBCharacteristic *)character error: (NSError *) error may be called back; it may be possible to determine whether data is sent successfully through error followed by corresponding operation. It may be also possible to send with the method WriteCommandWithData:callblock: and determine whether call back is successful through block.

4. Usage of wifi management

For single connection, the singletion method [WIFIManager shareWifiManager] is used to create the connection object, set the proxy and implement the proxy method.

ConnectWithHost: port: completion: is the connection method. It is possible to designate IP and port No., and determine whether callback is successful through block. It may be also possible to implement the proxy method-(void)WIFIManager:(WIFIManager *)manager didConnectedToHost:(NSString *)host port:(UInt16)port; to deal with the operations following connection success or failure.

```
@protocol XYWIFIManagerDelegate <NSObject>

// After successful connection to the host
- (void)WIFIManager:(WIFIManager *)manager
didConnectedToHost:(NSString *)host port:(UInt16)port;
// disconnect
- (void)XYWIFIManager:(WIFIManager *)manager
willDisconnectWithError:(NSError *)error;
// Execute after writing data
- (void)XYWIFIManager:(XYWIFIManager *)manager
didWriteDataWithTag:(long)tag;
// Method for performing after receiving data method
- (void)XYWIFIManager:(XYWIFIManager *)manager didReadData:(NSData
*)data tag:(long)tag;
// Method of execution after disconnection
- (void)XYWIFIManagerDidDisconnected:(XYWIFIManager *)manager;
@end
```