

```
////  
//// Anemometer iOS SDK Manual  
////
```

I. Functions

```
// Start scanning the BLE device.  
- (void) StartScan;  
  
// Stop scanning the BLE device  
- (void) StopScan;  
  
// Get the BLE device information. The peri and BTInfo are defined in the  
"BTInfo.h".  
- (BTInfo*) getBTDevice:(CBPeripheral*)peri;  
- (BTInfo*) getBTDeviceByIndex:(int)index;  
  
// Get the device index of the BLE device list.  
- (int) getBTDeviceIndex:(CBPeripheral*)peri;  
  
// Connect the BLE device.  
- (Boolean) connectBleDevice:(int)index;  
  
// Disconnect the BLE device.  
- (Boolean) disconnectBleDevice:(int)index;  
  
// Send data to the BLE device.  
// index: the index value of the BLE device.  
// data: the data sent to the anemometer.  
// lenght: the data length.  
- (Boolean) sendDataToBleById:(int)index buffer:(Byte*)data lenght:  
(int)lenght;  
  
// Set the UUID of the BLE device including service, read, and write.  
- (void) setUserUuid:(NSString*)service readUuid:(NSString*)readUuid  
writeUuid:(NSString*)writeUuid;  
  
// Get the BLE device list array.  
- (NSMutableArray*) getBTDevArrays;
```

II. Listening Events

```
// If the BLE interface of the smart phone completes the transmission of the  
data required to be sent to anemometer, the event onWriteDataOk is  
triggered and provides the transmitted data.  
- (void)onWriteDataOk:(DevData*)data {}  
  
// If the data from the anemometer (meter->app) is received in the BLE  
connection mode, the event onRecieveData is triggered and provides the
```

data received from the anemometer. For example, if the data “0x73” is sent to anemometer, the data “0x73 0x00 0x00 0x01 0x2b 0xf5 0xff 0xff 0x04 0xd2” can be received from anemometer.

- (void)onRecieveData:(DevData*)data {}

//If a new BLE device is found or the state of the existing BLE device is updated, the event **onBTConnectState** is triggered and provides the updated state.

// state == 0: Disconnect

// state == 1: Ready

// state == 2: Connecting

// state == 3: Fail Connection

// state == 4: Connected

- (void)onBTConnectState:(CBPeripheral*)peri state:(int)state; {}

// If a new BLE device is found or the broadcast data of any BLE device is updated in the broadcast mode, the event **onFoundNewBleDev** is triggered and provides the BLE device information. The info is defined in the “ScanInfo.h”.

- (void)onFoundNewBleDev:(ScanInfo*)info {}

// If the BLE device is connected, the event **onBTMacInfo** is triggered and provides the device data which the address of the BLE device can be analyzed from.

// data: the device data of the BLE device.

- (void)onBTMacInfo:(DevData*)data {}