

iOS Bluetooth Device SDK

Instruction Manual

Version 1.0.0 beta12

Catalog

1.Version Information.....	1
1.1. Version record.....	1
1.0.0. Version1.0.0	1
2. Operating environment	2
2.1. System requirements	2
2.2. Project settings	2
3.Interfaces	3
3.1. Definition	3
3.1.1. defaultPlugin	3
3.1.2. initPlugin	3
3.1.3. versionName	3
3.1.4. isBluetoothPowerOn	3
3.1.5. checkingBluetoothStatus	3
3.1.6. searchDevice	3
3.1.7. stopSearch.....	4
3.1.8. setDevices	4
3.1.9. addDevice.....	4
3.1.10. removeDevice	4
3.1.11. startAutoConnect.....	4
3.1.12. stopAutoConnect	4
3.1.13. checkConnectState	5
3.1.14. pushSetting	5
3.1.15. saveDebugMessage	5
3.1.16. appendlog	5
4.Callback.....	6
4.1. SearchResultsBlock	6
4.1.1. SearchResultsBlock	6
4.2. IBPushRespBlock	6
4.2.1. IBPushRespBlock.....	6
4.3. AHDeviceDataDelegate.....	7
4.3.1. didConnectStateChanged	7
4.3.2. didDataUpdateNotification	7
5. Instructions	8
5.1. Init.....	8
5.2. Device Scan	8
5.3. Device Connect & Data Sync	9
5.4. Device Settings.....	10
5.5.1 Temperature measurement mode settings.....	10
5.5.2 Thermometer historical data synchronization	10
5.5.3 Thermometer time synchronization.....	10
5.5.4 Thermometer start measuring	10
5.5.5 Thermometer data remove	11
5.5.6 Thermometer status sync settings	11

5.5.7 Blood pressure offline data sync settings.....	11
5.5.8 Blood pressure meter status sync settings.....	11
5.5.9 Blood pressure meter get device's sn.....	11
5.5.10 Blood pressure meter power off	11
5.5.11 Blood pressure data remove	12
5.5.12 Blood pressure meter user switching.....	12
5.5.13 Blood pressure meter voice control	12
5.5.14 Blood pressure meter manual measurement	12
5.6. Device data	13
5.6.1 Thermometer Data	13
5.6.2 Pulse Oximeter Data	13
5.6.3 Blood Pressure Meter Data.....	13

1. Version Information

1.1. Version record

1.0.0. Version1.0.0

Add basic function implementation and interface support for product AOJ-20A、AOJ-30B、AOJ-70B

2. Operating environment

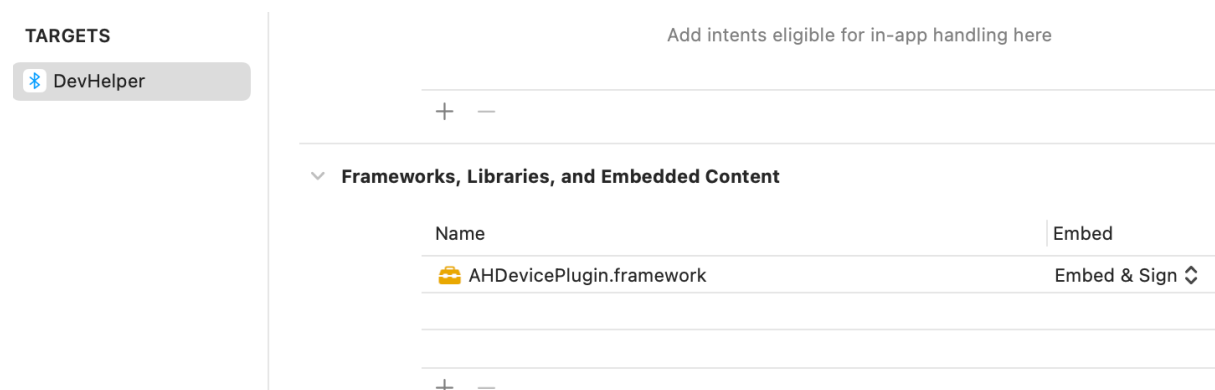
2.1. System requirements

Requires iOS 8.0 or above, and mobile phone hardware must support Bluetooth 4.0 or above.

2.2. Project settings

1、 import AHDevicePlugin.framework,

Set up “Frameworks,Libraries,and Embedded Content ”, Add dependency, set Embed to Embed & Sign



2、 import AHDevicePlugin-Bridging.h,set up "Swift Compiler- General" -> Objective-C Bridging Header, as follows \$(SRCROOT)/BluetoothDemo/AHDevicePlugin-Bridging.h

3、 Add Other Linker Flags, -all_load , -ObjC

4、 Update info.plist ,Add Privacy - Bluetooth Always Usage Description

5、 Update info.plist ,Add Required background modes

3.Interfaces

3.1. Definition

3.1.1. defaultPlugin

Static method, get the AHDevicePlugin object instance.

Parameter: Void

Return: AHDevicePlugin.

3.1.2. initPlugin

When using the object instance of AHDevicePlugin for the first time, you must first call this interface to initialize the SDK. All functions can be used normally after initialization

Parameter: dispatch_queue_t queue

Return: boolean

3.1.3. versionName

Get the version name of the SDK.

Parameter: Void

Return: String.

3.1.4. isBluetoothPowerOn

Check if the mobile phone's Bluetooth is turned on.

Parameter: Void

Return: boolean.

3.1.5. checkingBluetoothStatus

Register mobile phone Bluetooth status change notification.

Parameter: id<AHBluetoothStatusDelegate> delegate

Return: Void

3.1.6. searchDevice

Search for nearby Bluetooth devices.

Parameter:

① BTScanFilter *filter,

② SearchResultsBlock block, search results block

Return: boolean.

3.1.7. stopSearch

Stop search for nearby Bluetooth devices.

Parameter: Void

Return: boolean.

3.1.8. setDevices

Reset or clear the SDK measurement device list.

Parameter: NSArray *list:

Return: boolean.

3.1.9. addDevice

Add a single measuring device.

Parameter: BTDeviceInfo *device.

Return: boolean.

3.1.10. removeDevice

Remove the specified device and disconnect.

Parameter: String broadcastId.

Return: boolean.

3.1.11. startAutoConnect

Start the automatic connection synchronization service of the measurement device, and the interface supports the automatic reconnection mechanism.

The App does not need to handle the disconnected reconnecting, and it is not recommended to call the interface frequently.

Parameter:

① id<AHDeviceDataDelegate > Device's Data Delegate

Return: boolean.

3.1.12. stopAutoConnect

Stop the automatic synchronization of device data and disconnect all established Bluetooth connections.

Parameter: None

Return: boolean.

3.1.13. checkConnectState

Get the connection status of the specified device.

Parameter:

①String broadcastId.

Return: BTConnectState.

3.1.14. pushSetting

Device setting information update method, such as temperature measurement mode switching, user switching, etc .

Parameter:

①BTDeviceSetting *setting, abstract class, please refer to the definition of the subclass

②BTDeviceInfo***device**,

③IBPushRespBlock respBlock,

Return: void

3.1.15. saveDebugMessage

Enable the SDK log file logging feature.

Parameter:

①boolean status

②String logPath, log file storage path, using the absolute path

Return: void

3.1.16. appendlog

In the log log file, write the custom log information

Parameter:

①String msg

Return: void

4.Callback

4.1. *SearchResultsBlock*

4.1.1. SearchResultsBlock

void SearchResultsBlock (BTDeviceInfo *device)

Scanning result callback

Parameter:

① BTDeviceInfo *device

Return: void

4.2. IBPushRespBlock

4.2.1. IBPushRespBlock

void IBPushRespBlock (BOOL state, NSInteger error, id data)

Push setting callback

Parameter:

① BOOL state , YES=Success, NO=Failure

② NSInteger error, refer to the definition of BTErrrorCode

③ id data Additional data

Return: void

4.3. AHDeviceDataDelegate.

4.3.1. didConnectStateChanged

-(void)bleDevice:(BTDeviceInfo *)device didConnectStateChanged:(BTConnectionState)state;

Device connection state changes callback

Parameter:

① BTDeviceInfo *device.

② BTConnectState state.

Return: void

4.3.2. didDataUpdateNotification

-(void)bleDevice:(BTDeviceInfo *)dev didDataUpdateNotification:(BTDeviceData *)obj

Device measurement data and status callback.

Parameter:

① BTDeviceInfo*device.

② BTDeviceData *obj

Return: void

5. Instructions

5.1. Init

1、 Import AHDevicePlugin.framework

2、 Perform initialization in AppDelegate.swift

```
func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions:
[UIApplication.LaunchOptionsKey: Any]?) -> Bool {
    // log version
    let versionName : String?=AHDevicePlugin.default()?.versionName;
    print("version name:\(versionName ?? "default Name")");

    //init
    AHDevicePlugin.default()?.initManager(withDispatch: DispatchQueue.init(label:
"bluetoothQueue"));

    return true
}
```

5.2. Device Scan

```
//scan all devices
AHDevicePlugin.default()?.searchDevice(nil, results: { (device) in
    //TODO
})

//scan filter with device type
let filter = BTScanFilter();
filter.scanTypes = [BTDeviceType.oximeter.rawValue,
                    BTDeviceType.thermometer.rawValue,
                    BTDeviceType.digitalThermometer.rawValue,
                    BTDeviceType.bloodPressureMeter.rawValue];
AHDevicePlugin.default()?.searchDevice(filter, results: { (device) in
    //TODO
})
```

5.3. Device Connect & Data Sync

1、add device

App needs to fill in the device MAC, BroadcastID, ProtocolType, DeviceType when adding devices. Reference examples are shown below

```
let device:BTDeviceInfo = BTDeviceInfo.init();
device.broadcastId = ?
device.macAddress = ?
device.deviceType = ?
```

```
AHDevicePlugin.default()?.addDevice(device);
```

2、check manager status

```
if(AHDevicePlugin.default()?.managerStatus == BTManagerState.syncing){
    // In the Syncing state, the App does not need to call startAutoConnect repeatedly, and does not need to process the reconnection after disconnection. There is an automatic reconnection mechanism inside the SDK, and the App does not need to process this state.
    Return ;
}
```

3、Implementing a AHDeviceDataDelegate

```
extension ConnectViewController : AHDeviceDataDelegate
{
    func bleDevice(_ device: BTDeviceInfo!, didConnectStateChange state: BTConnectionState) {
        //handle connect state changed
    }

    func bleDevice(_ device: BTDeviceInfo!, didDataUpdateNotification obj: BTDeviceData!) {
        //handle device data
    }
}
```

4、start device auto connect & listening data

```
AHDevicePlugin.getInstance().startAutoConnect (listener);
```

5、stop device auto connect & listening data

```
AHDevicePlugin.getInstance().stopAutoConnect ();
```

5.4. Device Settings

All device settings, or data query, data deletion use the same method with different parameters

```
/**
 * setting, abstract class
 * different settings need to initialize specific instance
 * objects
 */
let setting :BTDeviceSetting;
let devic : BTDeviceInfo ;
AHDevicePlugin.default().push(setting, toDevice: device) { state,
error, data in
}
```

5.5.1 Temperature measurement mode settings

Modify the thermometer temperature measurement mode or unit, examples are as follows

```
let setting= AHTempSetting.init(cmd: AHTempCmd.configMode);

/**
 * Adult(1), Children(2), Ear(3), Material(4);
 */
setting.mode=AHTempMode.children;
```

5.5.2 Thermometer historical data synchronization

New Firmware Device

```
let setting = AHTempSetting.init(cmd:AHTempCmd.NewSyncData);
```

Old Firmware Device

```
let setting = AHTempSetting.init(cmd:AHTempCmd.SyncData);
```

5.5.3 Thermometer time synchronization

New Firmware Device

```
let setting = AHTempSetting.init(cmd:AHTempCmd.NewSyncTime);
```

Old Firmware Device

```
Let setting = AHTempSetting.init(cmd:AHTempCmd.SyncTime);
```

5.5.4 Thermometer start measuring

```
let setting = AHTempSetting.init(cmd:AHTempCmd.NewStartMeasuring);
```

5.5.5 Thermometer data remove

```
let setting = AHTempSetting.init(cmd:AHTempCmd.ClearData);
```

5.5.6 Thermometer status sync settings

```
let setting = AHTempSetting.init(cmd:AHTempCmd.QueryStatus);
```

5.5.7 Blood pressure offline data sync settings

Sync offline data based on device user

```
/**
 * 1 = Device User 1
 * 2 = Device User 2
 */
let setting = AHBpmSyncSetting.init();
setting.userNum = 1;
```

Sync all data

```
/**
 * true = Get all user data
 */
let setting = AHBpmSyncSetting.init();
setting.syncAll = true;
```

5.5.8 Blood pressure meter status sync settings

```
let setting= AHBpmConfigSetting.init()
setting.config = AHBpmConfig.statusSync;
```

5.5.9 Blood pressure meter get device's sn

```
let setting= AHBpmConfigSetting.init()
setting.config = AHBpmConfig.getSn;
```

5.5.10 Blood pressure meter power off

```
let setting= AHBpmConfigSetting.init()
setting.config = AHBpmConfig.powerOff;
```

5.5.11 Blood pressure data remove

Remove data based on device user

```
/**
 * 1 = Device User 1
 * 2 = Device User 2
 */
let setting = AHBpmSyncSetting.init();
setting.userNum = 1;
```

Remove all user data

```
/**
 * true = Remove all user data
 */
let setting = AHBpmRemoveSetting.init();
setting.removeAll = true;
```

5.5.12 Blood pressure meter user switching

```
/**
 * 1 = Device User 1
 * 2 = Device User 2
 */
let setting = new AHBpmConfigSetting.init();
setting.config = AHBpmConfig.switchUser;
setting.user = 1;
```

5.5.13 Blood pressure meter voice control

Turn on voice control

```
let setting = AHBpmConfigSetting.init();
setting.config = AHBpmConfig.VoiceControl;
setting.voiceState = true;
```

Turn off voice control

```
let setting = AHBpmConfigSetting.init();
setting.config = AHBpmConfig.VoiceControl;
setting.voiceState = false;
```

5.5.14 Blood pressure meter manual measurement

Start measuring

```
let setting = AHBpmConfigSetting.init();
setting.config = AHBpmConfig.StartMeasuring;
```

Stop measuring

```
let setting = AHBpmConfigSetting.init();
setting.config = AHBpmConfig.StopMeasuring;
```

5.6. Device data

5.6.1 Thermometer Data

1. AHTempStatus
<doc-zh/html/Classes/AHTempStatus.html>
2. AHTempData
<doc-zh/html/Classes/AHTempData.html>
3. AHTempSummary
<doc-zh/html/Classes/AHTempDataSummary.html>
4. AHTempModeData
<doc-zh/html/Classes/AHTempModeData.html>
5. AHTempErrorData
<doc-zh/html/Classes/AHTempErrorData.html>

5.6.2 Pulse Oximeter Data

1. AHSpO2
<doc-zh/html/Classes/AHSpO2.html>
2. AHPléthysmogram
<doc-zh/html/Classes/AHPléthysmogram.html>

5.6.3 Blood Pressure Meter Data

1. AHBpmData
<doc-zh/html/Classes/AHBpmData.html>
2. AHBpmProcessData
<doc-zh/html/Classes/AHBpmProcessData.html>
3. AHBpmConfigData
<doc-zh/html/Classes/AHBpmConfigData.html>
4. AHBpmStatus
<doc-zh/html/Classes/AHBpmStatus.html>