

Demo APP for MicroLifeDeviceSDK (WatchBP O3) - iOS

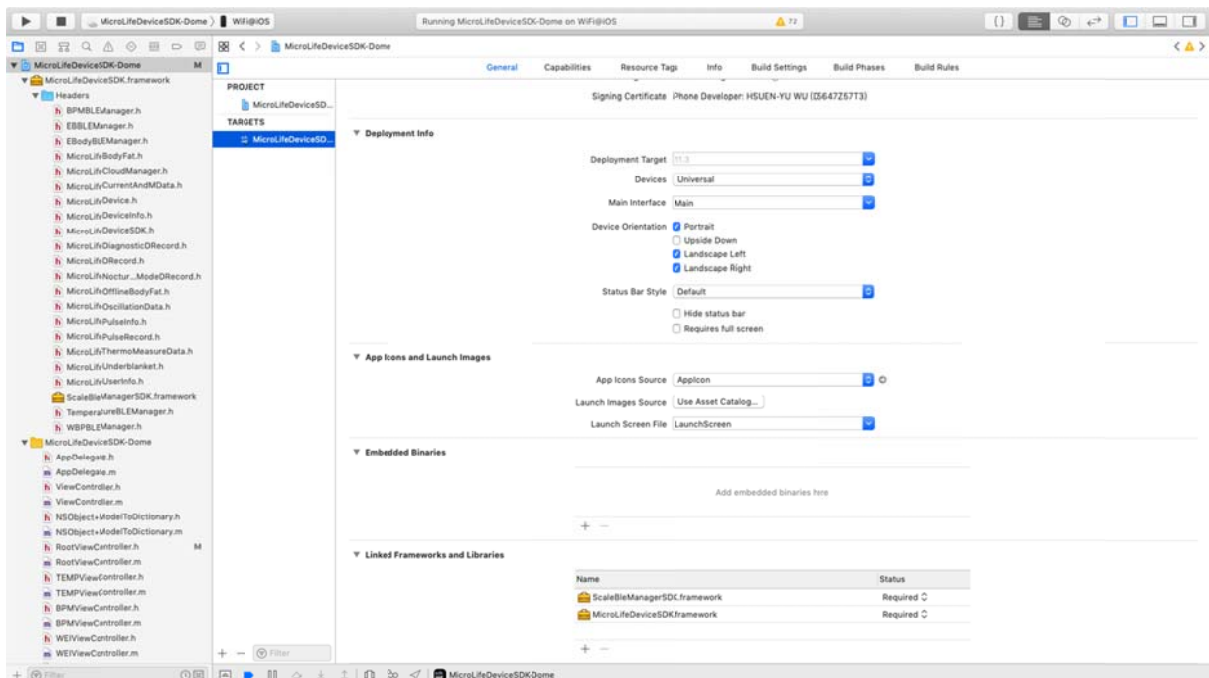
Table of Contents

- Chapter 1 Development Environment**
- Chapter 2 Entry Point and Bluetooth LE Protocol
 (BLEManager)**
- Chapter 3 APIs of BLEManager**
- Chapter 4 WatchBP O3 APIs**
- Chapter 5 User Interface & Functionality of WatchBP O3**
- Chapter 6 Demo App**

Chapter 1 Development Environment

This user manual serves as a quick introduction to MicroLifeDeviceSDK / APIs and shows how to integrate into an iOS Demo App.

- 1.1. First of all, add MicroLifeDeviceSDK.framework into a development project. The minimum supported version is iOS 10. The compatible version of Xcode IDE is 10.2.1.
- 1.2. Under TARGETS / General / Linked Frameworks and Libraries, add MicroLifeDeviceSDK.framework.



- 1.3. Under TARGETS / Build Settings / Enable, set Bitcode to NO.
- 1.4. Under TARGETS / Build Settings / Other Linker Flags, use the linker flag “-all_load”.
- 1.5. Import header file as bellows.

```
#import <MicroLifeDeviceSDK/MicroLifeDeviceSDK.h>
```

Chapter 2 Entry Point and Bluetooth LE Protocol (BLEManager)

The BLEManager is responsible for managing the Bluetooth communication. Refer to the file

“MicroLifeDeviceSDK/MicroLifeDeviceSDK.h” for suitable device models.

They are named like “XXXBLEManager”.



- 2.1. Initiate a BLEManager and register the delegation of BLEManager. The `DataResponseDelegate` is dedicated for response messages of BLEManager.
- 2.2. Once the BLEManager is initiated, the BLEManager checks the status of Bluetooth by `BLEManagerCellPhoneBluetoothDidUpdateState`.
- 2.3. While turning on Bluetooth, BLEManager runs the scan/discovery procedure automatically by `BLEManagerDidDiscoverBluetoothDeviceMacAddress` to get devices' information in the vicinity.
- 2.4. Call the `bindingDevice:(NSData *)macAddress` to bind an activated device. The status of binding can be found by the following section 2.5. The BLEManager will automatically connect to a bound device while it is activated.
- 2.5. Connection status :
 - 2.5.1. Connection / binding :

- (void) * BLEManagerDidConnectDevice;

2.5.2. Fail of connection :

- (void) * BLEManagerDidFailToConnectDevice;

2.5.3. Disconnection :

- (void) * BLEManagerDidDisconnectDevice;

Chapter 3 APIs of BLEManager

By utilizing the APIs of BLEManager, can transfer data between device and App via Bluetooth. The APIs includes two parts: one is for fundamental Bluetooth and another is dedicated for communication with a designated device / BPM (Blood Pressure Monitor) with specific commands and protocol.

3.1. Device searching :

3.1.1.

	<code>+(instancetype)shareInstanceWhithAuthorizationkey:(NSString *)key Target * Names:(NSArray *)target * Names</code>
Definition	Searching for device with default / customized name
Parameter	Key: Authorization code target * Names: device name which can be included single or multiple
	<pre>self.aBPMBLEManager = [BPMBLEManager shareInstanceWhithAuthorizationkey:\$DKkey TargetBPMNames:@[@"A6 BT",@"A6 BASIS PLUS BT",@"A7 TCUCH BT",@"B3 BT",@"B6 Connect",@"A6",@"Progress"]]; self.aBPMBLEManager.dataResponseDelegate = self;</pre>

3.1.2.

	<code>-(void) *</code> <code>BLEManagerCellPhoneBluetoothDidUpdateState:(MicroLifeBLEState)state;</code>
Definition	Response for the shareInstanceWhithAuthorizationkey (). This is to manage Bluetooth status of cellphone.
Parameter	status: Bluetooth status of cellphone

	<pre> * @enum MicroLifeBLEState * * @discussion Represents the current state of a BLE. * * @constant MicroLifeBLEStateUnknown=CBManagerStateUnknown State unknown, update imminent. * @constant MicroLifeBLEStateResetting=CBManagerStateResetting The connection with the system service was momentarily lost, update imminent. * @constant MicroLifeBLEStateUnsupported=CBManagerStateUnsupported The platform doesn't support the Bluetooth Low Energy Central/Client role. * @constant MicroLifeBLEStateUnauthorized=CBManagerStateUnauthorized The application is not authorized to use the Bluetooth Low Energy role. * @constant MicroLifeBLEStatePoweredOff=CBManagerStatePoweredOff Bluetooth is currently powered off. * @constant MicroLifeBLEStatePoweredOn=CBManagerStatePoweredOn Bluetooth is currently powered on and available to use. * */ typedef NS_ENUM(NSUInteger, MicroLifeBLEState) { MicroLife3LEStateUnknown = 0, MicroLife3LEStateResetting, MicroLife3LEStateUnsupported, MicroLife3LEStateUnauthorized, MicroLife3LEStatePoweredOff, MicroLife3LEStatePoweredOn, } </pre>
--	---

	<p>- (void) *</p> <p>BLEManagerDidDiscoverBluetoothDeviceMacAddress:(N SData *)macAddress Name:(NSString *)name RSSI:(NSNumber *)RSSI;</p>
Definition	<p>Response for the sharedInstanceWhithAuthorizationkey ().</p> <p>This is to manage information of devices that are discovered in the vicinity.</p>
Parameter	<p>macAddress: MAC address of device</p> <p>name: device name</p> <p>RSSI: RSSI</p>

3.2. Binding / Pairing :

3.2.1.

	- (void)bindingDevice:(NSData *)macAddress
Definition	Binding a specified device by MAC
Parameter	macAddress: MAC address of device

3.2.2.

	- (void) * BLEManagerDidConnectDevice;
Definition	Response for the bindingDevice() with the status Connection

	- (void) * BLEManagerDidFailToConnectDevice
Definition	Response for the bindingDevice() with the status Fail of Connection

	- (void) * BLEManagerDidDisconnectDevice;
Definition	Response for the bindingDevice() with the status Disconnection

3.3. MAC address of binding device :

3.3.1.

	- (NSData *)getBindingDevice;
Definition	Get MAC address of a binding device or nil.

3.4. Unbinding :

3.4.1.

	- (void)unBindingDevice;
Definition	Unbinding device

3.4.2.

	- (void) * BLEManagerDidFailToConnectDevice;
Definition	Response for the unBindingDevice()

3.5. Disconnection :

3.5.1.

	- (void)disconnectDevice;
Definition	Disconnect with device

3.5.2.

	- (void) * BLEManagerDidFailToConnectDevice;
Definition	Response for the disconnectDevice ()

3.6. Searching for devices :

3.6.1.

	- (void)reScan;
Definition	Search for devices repeatedly

3.6.2.

	- (void) * BLEManagerDidDiscoverBluetoothDeviceMacAddress:(NSData *)macAddress Name:(NSString *)name RSSI:(NSNumber *)RSSI;
Definition	This is to get information of devices which are discovered in the vicinity.
Parameter	macAddress: MAC of device name: device name RSSI: RSSI

3.7. Stop searching for devices :

3.7.1.

	- (void)stopScan;
Definition	Stop searching

3.7.2. Null.

Chapter 4 WatchBP O3 APIs

4.1. Read all history or current data :

4.1.1. Interface :

	-(void)readAllHistorys;
Definition	Read all history or current data

4.1.2. Delegate :

	- (void)WBO3BLEManagerResponseReadAllHistorys:(MicroLifeC DRecord *)data;
Parameter	data : History data

4.2. Read central BP memory data by index :

4.2.1. Interface :

	- (void)readCBPDataWithIndex:(int)index Dformat:(Dformat)dformat;
Definition	4.2. Read central BP memory data by index
Parameter	index : CBP memory index dformat : Data format NoCBPRaw : No CBP raw data LowCBPRaw : low resolution CBP data FullCBPRaw : full CBP raw data

4.2.2. Delegate :

	- (void)WBO3BLEManagerResponseReadCBPData:(MicroLifeC BPdataAndCalCBP *)data IsNoData:(BOOL)isNoData;
Parameter	cRecord : CBP data and CalCBP data isNullData : True or False

4.3. Clear all history data :

4.3.1. Interface :

	- (void)clearAllHistorys;
Definition	Clear all history data

4.3.2. Delegate :

	- (void)WBO3BLEManagerResponseClearHistory:(BOOL)isSuccess;
Parameter	isSuccess : True or False

4.4. Disconnect the Bluetooth device :

4.4.1. Interface :

	- (void)disconnect;
Definition	Disconnect the Bluetooth device

4.4.2. Delegate :

	- (void)WBO3BLEManagerDidDisconnectDevice;
--	--

4.5. Read user ID and version data

4.5.1. Interface :

	- (void)readUserAndVersionData;
Definition	Read user ID and version data

4.5.2. Delegate :

	- (void)WBO3BLEManagerResponseReadUserAndVersionData:(MicroLifeUserInfo *)user VersionData:(MicroLifeDeviceInfo *)verData;
Parameter	user : user ID

	verData : version data
--	------------------------

4.6. Write user ID :

4.6.1. Interface :

	- (void)writeUserID:(NSString *)ID;
Definition	Write user ID to device
Parameter	ID : User ID。

4.6.2. Delegate :

	- (void)WBO3BLEManagerResponseWriteUserID:(BOOL)isSuccess;
Parameter	isSuccess : True or False

4.7. Read ABPM setting values :

4.7.1. Interface :

	- (void)readSettingValues;
Definition	Read ABPM setting values

4.7.2. Delegate :

	- (void)WBO3BLEManagerResponseReadSettingValue:(MicroLifeSettingValues *)settingValues;
Parameter	settingValues : ABPM setting values

4.8. Write ABPM setting values :

4.8.1. Interface :

	- (void)writeSettingValues:(MicroLifeSettingValues *)settingValues;
Definition	Write ABPM setting values to device
Parameter	settingValues : ABPM setting values

4.8.2. Delegate :

	- (void)WBO3BLEManagerResponseWriteSettingValues:(BOOL) isSuccess;
Parameter	isSuccess : True or False

4.9. Read device ID and info :

4.9.1. Interface :

	- (void)readDeviceIDAndInfo;
Definition	Read device ID and info

4.9.2. Delegate :

	- (void)WBO3BLEManagerResponseReadDeviceIDAndInfo:(MicroLifeDeviceInfo *)deviceInfo;
Parameter	deviceInfo : Device ID and info

4.10. Read device time :

4.10.1. Interface :

	- (void)readDeviceTime;
Definition	Read device time

4.10.2. Delegate :

	- (void)WBO3BLEManagerResponseReadDeviceTime:(MicroLifeDeviceInfo *)deviceInfo;
Parameter	deviceInfo : Device Time

4.11. Write device Time

4.11.1. Interface :

	- (void)writeDeviceTime;
Definition	Write device Time to device

4.11.2. Delegate :

	- (void)WBO3BLEManagerResponseWriteDeviceTime:(BOOL)isSuccess;
Parameter	isSuccess : True or False

4.12. Read BPM function setting value :

4.12.1. Interface :

	- (void)readFunctionSettingValue;
Definition	Read BPM function setting value

4.12.2. Delegate :

	- (void)WBO3BLEManagerResponseReadFunctionSettingValue:(MicroLifeFunctionSettingValues *)functionSettingValues;
Parameter	functionSettingValues : BPM function setting value

4.13. Read BT module name :

4.13.1. Interface :

	- (void)readBTModuleName;
Definition	Read BT module name of device

4.13.2. Delegate :

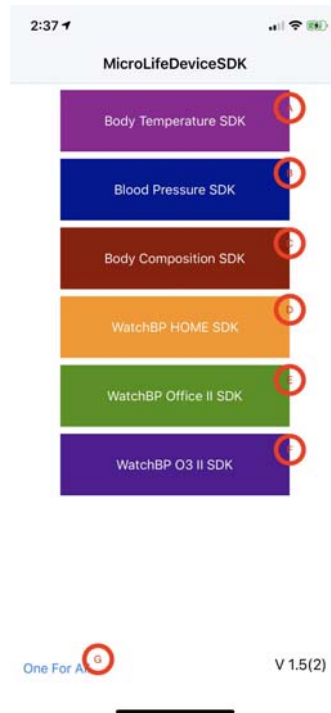
	- (void)WBO3BLEManagerResponseReadBTModuleName:(NSString *)BTModuleName;
Parameter	BTModuleName : BT Module Name

Chapter 5 User Interface & Functionality of WatchBP O3

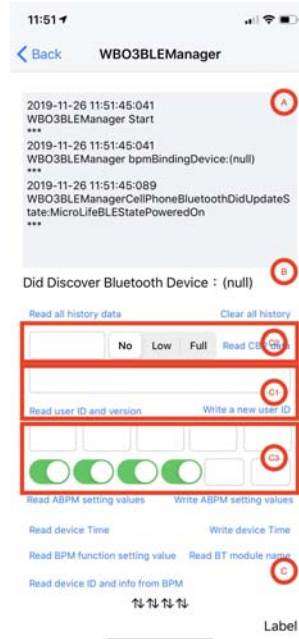
5.1. Getting Started :

Start the app and then select the button “WatchBP O3 II SDK” /

“F” to communicate with the designate device.



5.2. Operating Interface and Sequence :



- 5.2.1. Region A : The log window is used to display information about communication handshake between App and device.
- 5.2.2. Region B : The display is for current device with MAC address.
- 5.2.3. Region C : This part is for all functionalities of WatchBP O3 device.

C1 : Read CBP data by different mode such as Low/ Full. Before running this, the function “Read all history data” shall be performed firstly.

C2 : Read or Write User ID.

C3 : Read/ Write ABPM setting values :

1. ABPMStart : The starting time of the first measurement time zone
2. ABPMEnd : The end time of the first measurement time zone

3. ABPMInt_first : The interval of the first measurement time zone
4. ABPMInt_second : The interval of the second measurement time zone
5. HI_infPressure : The highest inflation pressure
6. CBP_zone2_meas_off : the second time zone of CBP measurement true:enabled/false:disabled
7. CBP_zone1_meas_off : the first time zone of CBP measurement true:enabled/false:disabled
8. SW_SEL_silent : Beeper
true:enabled/false:disabled
9. SW_checkhide : Hide(true)/Show(false) readings after measurement
10. CBPInt_first : The interval of the CBP first measurement time zone. Note: CBPInt_first should multiple times than ABPMInt_first.
11. CBPInt_second : The interval of the CBP second measurement time zone.

Note: CBPInt_second should multiple times than ABPMInt_second.

For instance, the specific CBP data can be read by an index number "330". The ABPM setting values is included 11 parameters "6, 23, 15, 30, 240, True, False, False, True, 20, 60":

The screenshot displays the MicroLifeDeviceSDK iOS app interface. At the top, there are two blue buttons: "Read all history data" and "Clear all history". Below these, a text input field contains "330", followed by three buttons: "No", "Low", and "Full". To the right of these buttons is a blue button labeled "Read CBP data". Below this is another text input field containing "XXXXXXXXXX".

The next section features two blue buttons: "Read user ID and version" and "Write a new user ID". Below these are five text input fields containing the values "6", "23", "15", "30", and "240". Below these fields are four toggle switches; the first and third are turned on (green), while the second and fourth are turned off (grey). To the right of the toggle switches are two text input fields containing "20" and "60".

Below the toggle switches are two blue buttons: "Read ABPM setting values" and "Write ABPM setting values". Below these are two blue buttons: "Read device Time" and "Write device Time". Below these are two blue buttons: "Read BPM function setting value" and "Read BT module name". Below these are two blue buttons: "Read device ID and info from BPM" and "Read device ID and info from BPM".

At the bottom, there are four blue buttons with the text "↕ ↕ ↕ ↕". Below these buttons is a blue button labeled "Label".

5.2.4. Refer to WBO3ViewController from the demo code (sample code) to get more detailed.

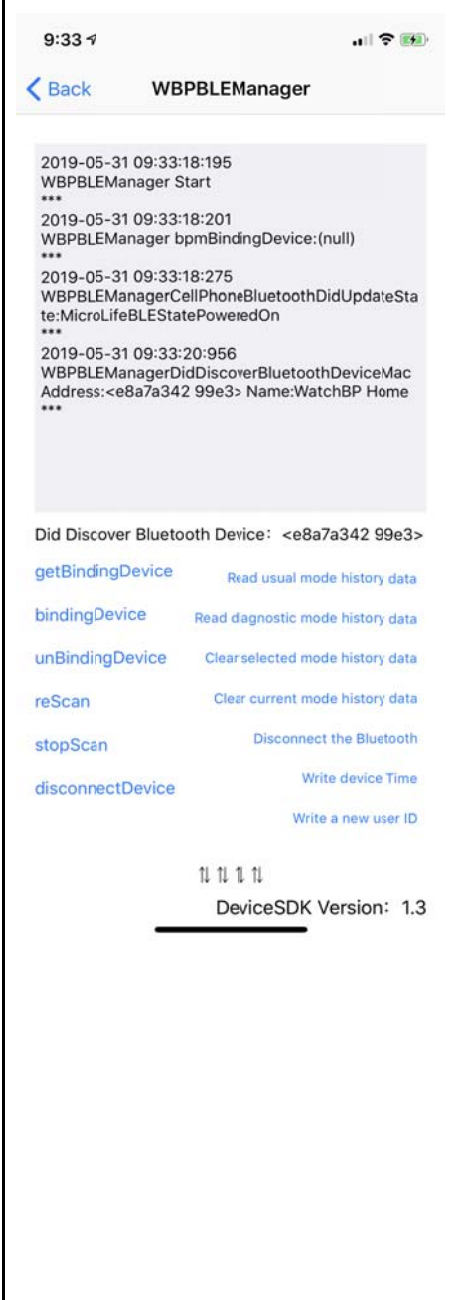
5.2.5. Operation Sequence :

- 5.2.5.1. The scanning (discovery) is automatically run to discover devices in the vicinity.
- 5.2.5.2. If a device is bonded, it will be connected accordingly. If not, the "bindingDevice" can be used to run bonding process.
- 5.2.5.3. Once the device is connected, select each function for communication with device.

Chapter 6 Demo App

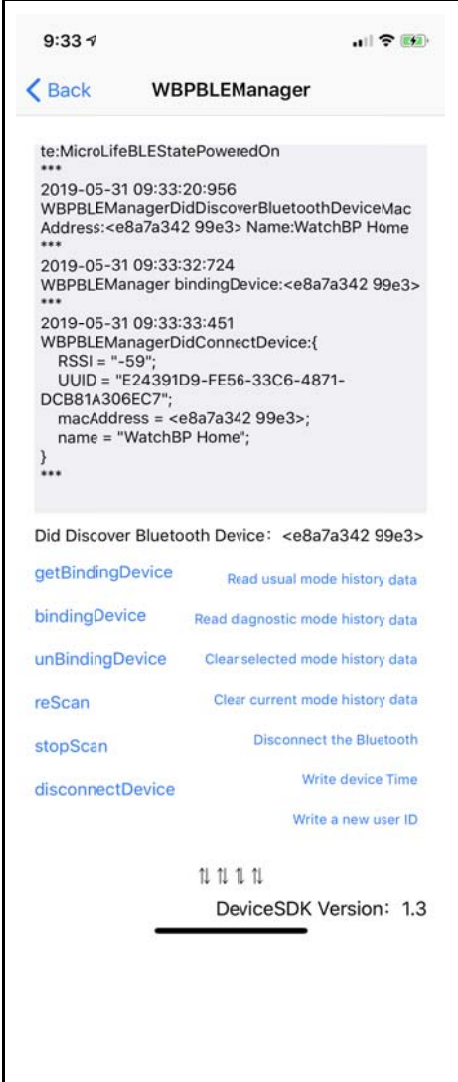
Each “XXXBLEManager” has the same operation sequences and features. The following demonstration is mainly conducted by using the model WBPBLEManager. Refer to the sample code to see the complete application for dedicated devices.

6.1. Scanning / Discovery :

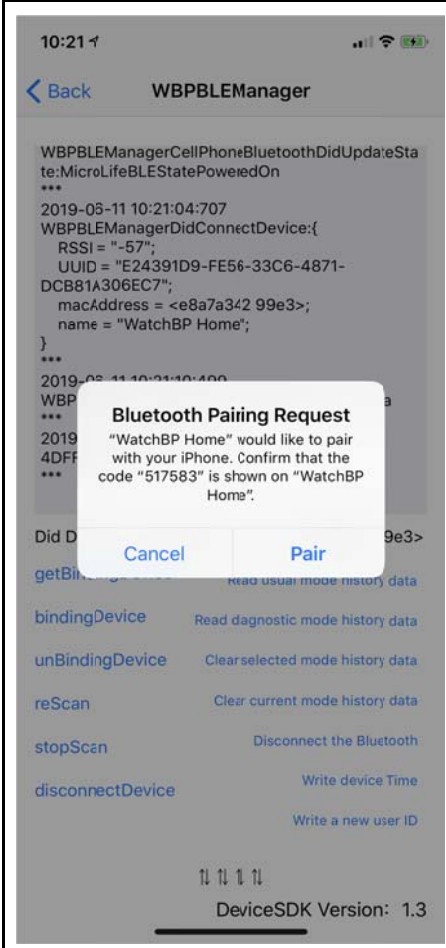
 <p>The screenshot shows the WBPBLEManager app interface. At the top, there's a status bar with the time 9:33 and signal indicators. Below the status bar is a navigation bar with a back arrow and the title 'WBPBLEManager'. The main content area displays a log of events with timestamps and messages. The log shows the app starting, the binding device being set to null, the Bluetooth state being updated to 'MicroLifeBLEStatePoweredOn', and a discovered Bluetooth device with MAC address <e8a7a342 99e3> and name 'WatchBP Home'. Below the log, there's a list of actions: getBindingDevice, bindingDevice, unBindingDevice, reScan, stopScan, and disconnectDevice. Each action has a corresponding description. At the bottom, there's a footer with the text 'DeviceSDK Version: 1.3'.</p>	<ol style="list-style-type: none"> 1. First of all, the WBPBLEManager starts Scanning / Discovery procedure. The detailed information will be displayed in Region A. 2. The WBPBLEManager bpmBindingDevice: * : (null) indicates that the remote device wasn't bonded yet. 3. The WBPBLEManagerCellPhoneBluetoothDidUpdateState:MicroLifeBLEStatePoweredOn indicates that the power status of Bluetooth from cell phone is On. 4. The WBPBLEManagerDidDiscoverBluetoothDeviceMacAddress:
---	--

	<p><e8a7a342 99e3> Name:WatchBP Home indicates that the device with MAC address <e8a7a342 99e3> is discovered nearby.</p> <p>5. In Region B, the “Did Discover Bluetooth Device : <e8a7a342 99e3>” stands for that the discovered/ selected device can be used in next steps such as Connection or Bonding.</p> <p>6. In Region C, click the function “bindingDevice” to run bonding process.</p>
--	---

6.2. Connection :

 <p>The screenshot shows the WBPBLEManager app interface. At the top, there's a status bar with the time 9:33 and signal indicators. Below the status bar is a navigation bar with a back arrow and the title 'WBPBLEManager'. The main content area displays a log of events: <ul style="list-style-type: none"> te:MicroLifeBLEStatePoweredOn *** 2019-05-31 09:33:20:956 WBPBLEManagerDidDiscoverBluetoothDeviceMac Address:<e8a7a342 99e3> Name:WatchBP Home *** 2019-05-31 09:33:32:724 WBPBLEManager bindingDevice:<e8a7a342 99e3> *** 2019-05-31 09:33:33:451 WBPBLEManagerDidConnectDevice:{ RSSI = "-59"; UUID = "E24391D9-FE56-33C6-4871-DCB81A306EC7"; macAddress = <e8a7a342 99e3>; name = "WatchBP Home"; } *** Below the log, there are several buttons with corresponding actions: <ul style="list-style-type: none"> getBindingDevice (Read usual mode history data) bindingDevice (Read diagnostic mode history data) unBindingDevice (Clear selected mode history data) reScan (Clear current mode history data) stopScan (Disconnect the Bluetooth) disconnectDevice (Write device Time) (Write a new user ID) At the bottom, there's a footer with the text 'DeviceSDK Version: 1.3'.</p>	<p>1. The information about the device bonding will be shown “WBPBLEManager bindingDevice:<e8a7a342 99e3>” in Region A.</p> <p>2. The following indicates that the device is connected. WBPBLEManagerDidConnect Device:{</p> <pre> RSSI = "-68"; UUID = "E24391D9-FE56-33C6-4871-DCB81A306EC7"; macAddress = <e8a7a342 99e3>; name = "WatchBP Home"; }</pre>
---	--

6.3. Bonding / Pairing :



The screenshot shows the WBPBLEManager app interface. At the top, there is a status bar with the time 10:21 and signal indicators. Below the status bar is a navigation bar with a back arrow and the title "WBPBLEManager". The main content area displays a log of events, including a timestamp "2019-03-11 10:21:04:707" and a dictionary of device information: {"RSSI": "-57", "UUID": "E24391D9-FE56-33C6-4871-DCB81A306EC7", "macAddress": "<e8a7a342 99e3>", "name": "WatchBP Home"}. A modal dialog titled "Bluetooth Pairing Request" is overlaid on the screen, asking the user to confirm pairing with "WatchBP Home" and showing the code "517583". The dialog has "Cancel" and "Pair" buttons. Below the dialog, there is a list of actions: "getBindingDevice" (Read usual mode history data), "bindingDevice" (Read diagnostic mode history data), "unBindingDevice" (Clear selected mode history data), "reScan" (Clear current mode history data), "stopScan" (Disconnect the Bluetooth), "disconnectDevice" (Write device Time), and "Write a new user ID". At the bottom, there is a footer with the text "DeviceSDK Version: 1.3".

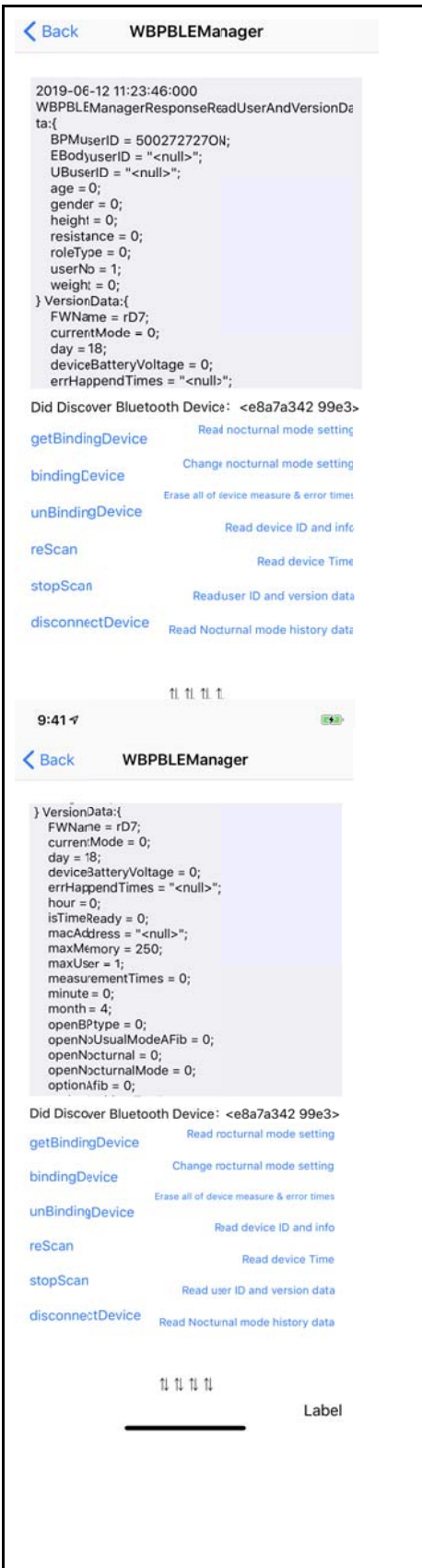
1. There is a message to confirm the bonding between device and cellphone if they haven't bonded yet.

2. Click "Pair" to run the bonding process.

6.4. Testing Command : Write a new user ID to BPM

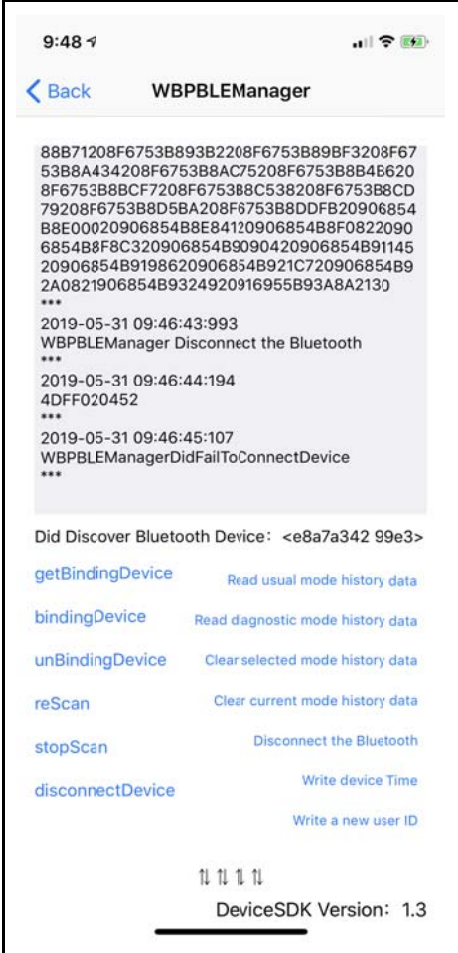
<div> < Back WBPBLEManager </div> <pre> 6854B8F8C320906854B9090420906854B91145 20906854B9198620906854B921C720906854B9 2A0821906854B9324920916955B93A8A213D *** 2019-06-11 10:24:46:405 WBPBLEManager Write a new user ID: 835702427AY *** 2019-06-11 10:24:46:603 4DFF18060038333537303234323741590000000 000000000000DA *** 2019-06-11 10:24:46:931 WBPBLEManagerResponseWriteUserID:YES *** 2019-06-11 10:24:47:110 4D5103068128 *** Did Discover Bluetooth Device: <e8a7a342 99e3> getBindingDevice Read usual mode history data bindingDevice Read diagnostic mode history data unBindingDevice Clear selected mode history data reScan Clear current mode history data stopScan Disconnect the Bluetooth disconnectDevice Write device Time Write a new user ID 11 11 11 11 </pre>	<ol style="list-style-type: none"> 1. The command “Write a new user ID” is to write a new user ID to device. 2. The following “WBPBLEManager Write a new user ID : 835702427AY” indicates that the writing value is “835702427AY” made up of ASCII code. 3. The response is as below: WBPBLEManagerResponse WriteUserID:YES. It means that the process is successful. 4. It can be checked by the command “Read user ID and version data”.
--	--

6.5. Testing Command : Read user ID and version data from BPM

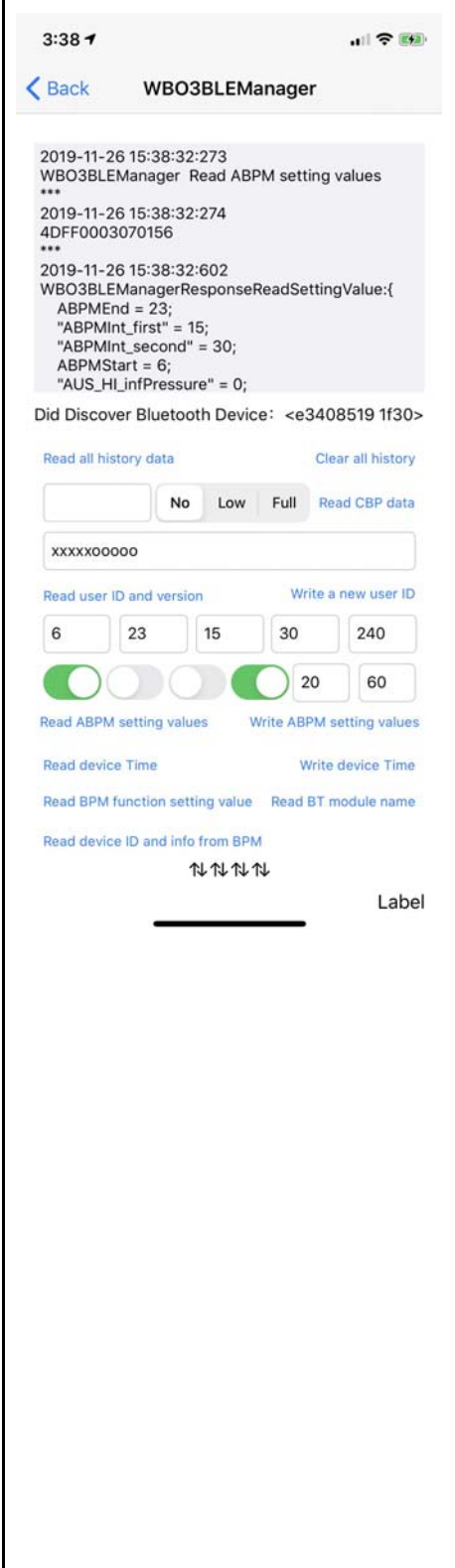
 <p>2019-06-12 11:23:46:000 WBPBLEManagerResponseReadUserAndVersionData: ta:{ BPMUserID = 500272727ON; EBodyuserID = "<null>"; UUserID = "<null>"; age = 0; gender = 0; height = 0; resistance = 0; roleType = 0; userNo = 1; weight = 0; } VersionData:{ FWName = rD7; currentMode = 0; day = 18; deviceBatteryVoltage = 0; errHappendTimes = "<null>"; }</p> <p>Did Discover Bluetooth Device: <e8a7a342 99e3> getBindingDevice Read nocturnal mode setting bindingDevice Change nocturnal mode setting unBindingDevice Erase all of device measure & error times reScan Read device ID and info stopScan Read device Time disconnectDevice Read user ID and version data disconnectDevice Read Nocturnal mode history data</p> <p>9:41</p> <p>WBPBLEManager</p> <p>VersionData:{ FWName = rD7; currentMode = 0; day = 18; deviceBatteryVoltage = 0; errHappendTimes = "<null>"; hour = 0; isTimeReady = 0; macAddress = "<null>"; maxMemory = 250; maxUser = 1; measurementTimes = 0; minute = 0; month = 4; openBPtype = 0; openNoUsualModeAFib = 0; openNocturnal = 0; openNocturnalMode = 0; optionAFib = 0; }</p> <p>Did Discover Bluetooth Device: <e8a7a342 99e3> getBindingDevice Read nocturnal mode setting bindingDevice Change nocturnal mode setting unBindingDevice Erase all of device measure & error times reScan Read device ID and info stopScan Read device Time disconnectDevice Read user ID and version data disconnectDevice Read Nocturnal mode history data</p> <p>Label</p>	<p>1. The command "Read user ID and version data" is to receive the information of user ID and device version.</p> <p>2. The response is as below: WBPBLEManagerResponseReadUserAndVersionData:{ BPMUserID = 500272727ON; EBodyuserID = "<null>"; UUserID = "<null>"; age = 0; gender = 0; height = 0; resistance = 0; roleType = 0; userNo = 1; weight = 0; } VersionData:{ FWName = rD7; currentMode = 0; day = 18; deviceBatteryVoltage = 0; errHappendTimes = "<null>"; hour = 0; isTimeReady = 0; macAddress = "<null>"; maxMemory = 250;</p>
--	--

	<pre>maxUser = 1; measurementTimes = 0; minute = 0; month = 4; openBPtype = 0; openNoUsualModeAFib = 0; openNocturnal = 0; openNocturnalMode = 0; optionAfib = 0; optionAmbientT = 1; optionDeviceID = 0; optionDiagnosticModeAFib = 0; optionIHB = 1; optionMAM = 1; optionTubeless = 0; protocolVersion = 0; second = 0; year = 2019; }</pre> <p>3. The “BPMuserID” stands for user ID and the value is “500272727ON”.</p> <p>4. The “VersionData” stands for device version included name of firmware “FWName” and its building date “day”, “month” and “year”.</p>
--	--

6.6. Disconnection

 <p>The screenshot shows the WBPBLEManager app interface. At the top, the status bar displays the time 9:48 and signal indicators. Below the title bar, there is a log of Bluetooth operations. The log includes a long hexadecimal string, a timestamp 2019-05-31 09:46:43:993, and the message 'WBPBLEManager Disconnect the Bluetooth'. Below this, another timestamp 2019-05-31 09:46:44:194 and the address 4DFF020452 are shown. Further down, a timestamp 2019-05-31 09:46:45:107 and the message 'WBPBLEManagerDidFailToConnectDevice' are displayed. Below the log, there is a message 'Did Discover Bluetooth Device: <e8a7a342 99e3>'. At the bottom, there is a list of actions with their corresponding descriptions: 'getBindingDevice' (Read usual mode history data), 'bindingDevice' (Read diagnostic mode history data), 'unBindingDevice' (Clear selected mode history data), 'reScan' (Clear current mode history data), 'stopScan' (Disconnect the Bluetooth), and 'disconnectDevice' (Write device Time). At the very bottom, the text 'DeviceSDK Version: 1.3' is displayed.</p>	<p>1. In order to disconnect the device completely, either “disconnectDevice” (Region C) or “Disconnect the Bluetooth” (Region D) is available.</p> <p>2. The response is the following “WBPBLEManagerDidFailToConnectDevice” once the disconnection command is executed.</p>
---	---

6.7. Testing Command : Read ABPM setting values (WBO3BLEManager)

 <p>The screenshot shows the WBO3BLEManager app interface. At the top, there's a status bar with the time 3:38 and signal/battery icons. Below the title bar, there's a log area showing three entries: a timestamp and command, a timestamp and device address, and a timestamp and a JSON response. Below the log, there are buttons for 'Read all history data' and 'Clear all history'. A section with a text input field and buttons 'No', 'Low', 'Full', and 'Read CBP data' is present. Below that is a section for 'Read user ID and version' with five input fields containing values 6, 23, 15, 30, and 240, and two toggle switches. Further down are buttons for 'Read ABPM setting values', 'Write ABPM setting values', 'Read device Time', 'Write device Time', 'Read BPM function setting value', 'Read BT module name', and 'Read device ID and info from BPM'. At the bottom, there's a label 'Label' next to a horizontal line.</p>	<p>1. The command “Read ABPM setting values” is to read the setting values of ABPM mode.</p> <p>2. The response is the following:</p> <pre>WBO3BLEManagerResponse ReadSettingValue :{ ABPMEnd = 23; "ABPMInt_first" = 15; "ABPMInt_second" = 30; ABPMStart = 6; "AUS_HI_infPressure" = 0; AutoMeasureNumber = 0; "CBPInt_first" = 20; "CBPInt_second" = 60; "CBP_zone1_meas_off" = 0; "CBP_zone2_meas_off" = 1; "HI_infPressure" = 240; IntervalTime = 0; RestTime = 0; "SW_AFib" = 0; "SW_AMPM" = 0; "SW_AUS_Hide" = 0; "SW_AUTO_hide" = 0;</pre>
--	--

	<pre>"SW_AVG_no_include_first" = 0; "SW_CBP" = 0; "SW_Kpa" = 0; "SW_SEL_silent" = 0; "SW_checkhide" = 1; }</pre>
--	--