





## Chapter

1. Environment Configuration Instructions
2. SDK Tools
3. SDK Bluetooth Interface Instructions
4. MicroLifeDevice Interface Instructions
5. Oxygen Interface Instructions
6. Use Process
7. Dome Code Instructions
8. Operation Instructions
9. Update Records

1. Environment Configuration Instructions:
  - 1.1. SDK Minimum Supported Version: iOS 13
  - 1.2. Drag MicroLifeDeviceSDK.framework into the project.
  - 1.3. Add MicroLifeDeviceSDK.framework to TARGETS / General / Frameworks, Libraries, and Embedded Content.
  - 1.4. To get the local Log, you need to add ZipArchive.framework in MicroLifeDeviceSDK/framework to frameworks, Libraries, and Embedded Content, and set "Embed & Sign".
  - 1.5. To use the body fat machine, you need to add the BelterScaleInfo.framework in MicroLifeDeviceSDK/framework to frameworks, Libraries, and Embedded Content, and set "Do Not Embed".

▼ Frameworks, Libraries, and Embedded Content

Name	Embed
 MessageUI.framework	Do Not Embed ⚙
 MicroLifeDeviceSDK.framework	Do Not Embed ⚙
 ScaleBleManagerSDK.framework	Do Not Embed ⚙
 ZipArchive.framework	Embed & Sign ⚙
+ —	

- 1.6. Add -all\_load in TARGETS / Build Settings / Other Linkeder Flags.
- 1.7. Where you need to use it, just import the header file.

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

## 2. SDK tool description:

## 2.1. Log Management: Local Log recording tool

```
//Log Management
#import <MicroLifeDeviceSDK/LogManager.h>
```

## 2.2. Bluetooth Core:

2.2.1. BLESDK: Bluetooth bottom manager, responsible for all Bluetooth general operations.

2.2.2. BLEDeviceInfo: Bluetooth device tool, Bluetooth scan response basic object.

2.2.3. MicroLifeDevice: MicroLife Device Tool. Responsible for the transmission and parsing of all kinds of equipment instructions of Paragon.

2.2.4. MicroLifeDataModel: MicroLife's general data object.

2.2.5. MicroLifeDeviceInfo: MicroLife device object.

2.2.6. MicroLifeUserInfo: MicroLife User Object.

```
//Bluetooth Core
#import <MicroLifeDeviceSDK/BLESDK.h>
#import <MicroLifeDeviceSDK/BLEDeviceInfo.h>
#import <MicroLifeDeviceSDK/MicroLifeDevice.h>
#import <MicroLifeDeviceSDK/MicroLifeDataModel.h>

//Data Model
#import <MicroLifeDeviceSDK/MicroLifeDeviceInfo.h>
#import <MicroLifeDeviceSDK/MicroLifeUserInfo.h>
```

## 2.3. Oxygen:

2.3.1. MicroLifeOxygen: Oxygen Device Tool.

2.3.2. MicroLifeOxygenData: Oxygen data.

```
//Oxygen
#import <MicroLifeDeviceSDK/MicroLifeOxygen.h>
#import <MicroLifeDeviceSDK/MicroLifeOxygenData.h>
```

## 3. SDK Bluetooth interface description:

## 3.1. Singleton:

	+ (instancetype)shareOne;
Definition	Singleton

## 3.2. instantiation:

	+ (instancetype)share;
Definition	instantiation

## 3.3. Show Log:

	- (void)showLog:(BOOL)showLog;
Definition	shows the SDK running Log
Parameter	showLog: Whether to display the Log

## 3.4. Set MicroLife Device:

	- (void)device:(NSArray *)bluetooth;
Definition	to set the MicroLife device, when the mobile phone's Bluetooth is turned on, it will automatically scan the surrounding devices, find the matching device, the default is automatic Connect the device.
Parameter	bluetooth: MicroLifeDevice Class
	<pre>[self.sdk     device:@     [self.bp3g,self.bp4g,self.temperature,self.oxygen,self     .weight,self.bloodSugar]];</pre>

## 3.5. Set Auto Scan:

	- (void)autoScan:(BOOL)autoScan;
Definition	When CBManagerStatePoweredOn automatically opens scanning [Default: Open]
Parameter	autoScan: Auto Scan

## 3.6. Set Scan stop Time:

	- (void)stopTime:(NSInteger) time;
Definition	Set the scan stop time
Parameter	time: Scan stop time

## 3.7. Set RSSI:

	- (void)rssi:(NSInteger)rssi;
Definition	scan RSSI strength
Parameter	rssi: Scan RSSI strength

## 3.8. Start Scan:

	- (void)startScan;
Definition	Start Scan

## 3.9. Cancel Scan:

	- (void)cancelScan;
Definition	Cancel Scan

## 3.10. Cancel All Connect:

	- (void)cancelAllConnect;
Definition	Cancel All Connect

## 3.11. Get the currently connected devices:

	- (NSArray *)findConnectedDevices;
Definition	Get the currently connected devices

## 3.12. When CentralManager state changed:

	- (void)getDidUpdateStateBlock:(didUpdateStateBlock)block;
DefinitionM onitor	Bluetooth state of the mobile
phonePara meter	void(^didUpdateStateBlock) (CBManagerState state)

## 3.13. Scan Device:

	- (void)getScanDeviceBlock:(scanDeviceBlock)scanDeviceBlock CancelScanBlock:(cancelScanBlock)cancelScanBlock;
DefinitionM onitor	scan device-like State
Parameter	void(^scanDeviceBlock) (id device)
Parameter	void(^cancelScanBlock) (void)

## 3.14. Connect Device State:

	- (void)getConnectDeviceStateBlock:(connectDeviceStateBlock) connectDeviceStateBlock CancelAllDevicesConnectionBlock:(cancelAllDevicesConnectionBlock) cancelAllDevicesConnectionBlock;
DefinitionM onitor	connected device
stateParam eter	void( ^connectDeviceStateBlock) (id device, CBCentralManager *central, CBPeripheral *peripheral, CBPeripheralState state, NSError * error)
Parameter	void(^cancelAllDevicesConnectionBlock) (void)

## 4. MicroLifeDevice interface description:

## 4.1. Singleton:

	+ (instancetype)shareWhithAuthorizationkey:(NSString *)key ;
Definition	Singleton
Parameter	key: Authorization code

## 4.2. Add Device Model:

	- (void)addDeviceModel:(NSArray *)models;
Definition	Search device custom model
Parameter	models: device name which can be included single or multiple
	<pre>[self.watchBP03 addDeviceModel:@[@"Watch BP 03"]];</pre>

## 4.3. Set Auto Scan:

	- (void)autoScan:(BOOL)autoScan;
Definition	When CBManagerStatePoweredOn automatically opens scanning [Default: Open]
Parameter	autoScan: Auto Scan

## 4.4. Auto Connect [Default: Open]:

	- (void)setAutoConnect:(BOOL)autoConnect;
Definition	Set Auto Connect [Default: Open]
Parameter	autoConnect: Auto Connect

## 4.5. ReConnect Time [Default: 0]:

	- (void)setReConnectTime:(NSInteger)reConnectIndex;
Definition	Set ReConnect Time [Default: 0],interval
Parameter	reConnectIndex: ReConnect Time

## 4.6. Set Scan stop Time:

	- (void)stopTime:(NSInteger)time;
Definition	Set the scan stop time, Bluetooth independent management mode uses
Parameter	time: Scan stop time

## 4.7. Set RSSI:

	- (void) rssi:(NSInteger)rssi;
Definition	Set the scan RSSI strength, use
Parameter	rssi: Scan RSSI strength

## 4.8. Start Scan:

	- (void)startScan;
Definition	Start Scan, Bluetooth independent management mode use

## 4.9. Cancel Scan:

	- (void )cancelScan;
--	----------------------

Definition	Cancel Scan, Bluetooth independent management mode uses
------------	---

## 4.10. Connect:

	- (void)connectDevice;
Definition	Connect

## 4.11. Disconnect:

	- (void)disconnectDevice;
Definition	Disconnect

## 4.12. Set Auto Reconnect:

	(void)setAutoReconnect
;	-Default start

## 4.13. Cancel Auto Reconnect:

	- (void)cancelAutoReconnect;
DefinitionCancel	automatic reconnection

## 4.14. Device Connected:

	- (void)deviceConnectedBlock:(deviceConnectedBlock)block;
DefinitionMonitor	device has registered all default services
Parameter	void(^deviceConnectedBlock) (void);

## 4.15. Device independent bluetooth:

	- (void)getDidUpdateStateBlock:(didUpdateStateBlock)didUpdateStateBlock ScanDeviceBlock:(scanDeviceBlock)scanDeviceBlock CancelScanBlock:(cancelScanBlock)cancelScanBlock ConnectDeviceStateBlock:(connectDeviceStateBlock)connectDeviceStateBlock CancelAllDevicesConnectionBlock:(cancelAllDevicesConnectionBlock)
cancelAllDevicesConnectionBlock	; Mobile phone Bluetooth status, device scan and connection status
Parameter	void(^didUpdateStateBlock) (CBManagerState state)
Parameter	void(^scanDeviceBlock) (id device)
Parameter	void(^cancelScanBlock) (void)
Parameter	void(^connectDeviceStateBlock) (id device, CBCentralManager *central, CBPeripheral *peripheral, CBPeripheralState state, NSError * error)
Parameter	void(^cancelAllDevicesConnectionBlock) (void)

## 4.16. Read RSSI:

	- (void)getReadRSSIBlock:(readRSSIBlock)block;
Definition	Monitor device RSSI status.
Parameter	typedef void(^readRSSIBlock) (NSNumber *RSSI, NSError * error);

## 4.17. Read Data Value:

	- (void)getReadDataValueBlock:(readDataValueBlock)block;
Definition	Monitor response status (original value).
Parameter	void(^readDataValueBlock) (NSInteger CMD, NSData * value, NSError * error);

## 4.18. Read Data Model:

	- (void)getReadDataModelBlock:(readDataModelBlock)block;
DefinitionMonitor	response status.
Parameter	void(^readDataModelBlock) (NSInteger CMD, id model, NSError * error);

## 4.19. Read Validation item Error:

	- (void)getValidationErrorBlock:(validationErrorBlock)block;
DefinitionMonitor	error response status.
Parameter	void(^validationErrorBlock) (NSString *item, NSString *info, NSData * value);



## 5. Oxygen interface description

- 5.1. Set monitoring response status, and obtain corresponding data according to CMD

```
[self.oxygen getReadDataModelBlock:^(NSInteger CMD, id _Nonnull
model, NSError * _Nonnull error) {
[weakself log:weakself.oxygen CMD:CMD Model:model Error:error];
switch (CMD) {
case CMDOxygenVolumeTracingData:
[weakself showOxygen:model];
break;
case CMDOxygenSaturationAndPulseRateData:
break;
case CMDOxygenSaturationAndPulseRateAlarmLimits:
break;
default:
break;
}
}];
```

## 5.2. Read Oxygen Volume Tracing Data:

Definition	Oxygen Volume Tracing Data
CMD	CMDOxygenVolumeTracingData
model	MicroLifeOxygenData
	<pre>case CMDOxygenVolumeTracingData: [weakself showOxygen:model]; break;</pre>

## 5.3. Read Oxygen Saturation And Pulse Rate Data:

Definition	Oxygen Saturation And Pulse Rate Data
CMD	OxygenSaturationAndPulseRateData
model	MicroLifeOxygenData
	<pre>case CMDOxygenSaturationAndPulseRateData: break;</pre>

## 5.4. Write blood oxygen saturation and pulse rate alarm limit:

	-(void)alarmSPOMax:(NSInteger)maxSPO SPOmin:(NSInteger)minSPO PlusMax:(NSInteger)maxPlus Plusmin:(NSInteger)minPlus;
Definition	Write blood oxygen saturation and pulse rate alarm limit
Parameter	maxSPO The lower limit of blood oxygen alarm limit, the value is 50-100. minSPO The upper limit of blood oxygen alarm limit, the value is 50- 100. maxPlus The lower limit of the pulse rate alarm limit, the value is 5-250. It must be a multiple of 5 and cannot be equal to 0.

	minPlus The upper limit of the pulse rate alarm limit, the value is 5- 250, must be a multiple of 5, and cannot be equal to 0.
CMD	CMDOxygenSaturationAndPulseRateAlarmLimits
model	MicroLifeDeviceInfo
	<pre>case CMDOxygenSaturationAndPulseRateAlarmLimits:     break;</pre>

## 6. usage process:

## 6.1. independent management mode (multiple devices are connected at the same time):

## 6.1.1. single instance of Parallax device:

```
self.oxygen = [MicroLifeOxygen
shareWhithAuthorizationkey:SDKkey_SP02];
```

## 6.1.2. monitor mobile phone Bluetooth status, device scanning and Connection status: When the Bluetooth of the mobile phone is turned on, it will automatically scan the surrounding devices, find a matching device, and automatically connect to the device by default. After the device is connected, if no command is sent for more than 120 seconds, it will automatically disconnect.

```
[self.oxygen getDidUpdateStateBlock:^(CBManagerState state) {
    NSString *log = [NSString stringWithFormat:@"oxygen
    DidUpdateState :%ld", (long)state];
    [weakself addLogWhitText:log];
} ScanDeviceBlock:^(id _Nonnull device) {
    NSString *log = [NSString stringWithFormat:@"oxygen ScanDevice
    Name :%@ UUID :%@ mac :%@", [device name], [device
    UUID], [device mac]];
    [weakself addLogWhitText:log];
} CancelScanBlock:^(
    [weakself addLogWhitText:@"oxygen Cancel Scan"];
    [weakself.oxygen startScan];
} ConnectDeviceStateBlock:^(id _Nonnull device, CBCentralManager
    * _Nonnull central, CBPeripheral * _Nonnull peripheral,
    CBPeripheralState state, NSError * _Nonnull error) {
    [weakself connectDeviceInfo:device PeripheralState:state];
} CancelAllDevicesConnectionBlock:^(
    [weakself addLogWhitText:@"oxygen Cancel All Devices
    Connection"];
    [weakself.oxygen startScan];
}];
```

## 6.1.3. Monitoring command response status: Send commands according to usage needs, and obtain corresponding information through CMD.

```
[self.oxygen getReadDataModelBlock:^(NSInteger CMD, id _Nonnull
    model, NSError * _Nonnull error) {
    [weakself log:weakself.oxygen CMD:CMD Model:model Error:error];
}];
```

## 6.1.4. Monitor error response status: When a response is received, it will automatically disconnect.

```
[self.oxygen getValidationErrorBlock:^(NSString * _Nonnull item,  
    NSString * _Nonnull info, NSData * _Nonnull value) {  
  
}];
```

## 6.2. Unified management mode (single device connection):

## 6.2.1. single instance Bluetooth manager:

```
// Bluetooth scanner establishment
self.sdk = [BLESDK shareOne];
```

## 6.2.2. single instance Parallax device:

```
self.oxygen = [MicroLifeOxygen
shareWhithAuthorizationkey:SDKkey_SP02];
```

## 6.2.3. monitor mobile phone Bluetooth status:

```
// Bluetooth status of mobile phone
[self.sdk getDidUpdateStateBlock:^(CBManagerState state) {
    NSString *log = [NSString
        stringWithFormat:@"DidUpdateState :%ld", (long)state];
    [weakself addLogWhitText:log];
}];
```

## 6.2.4. monitor device scan status:

```
[self.sdk getScanDeviceBlock:^(id _Nonnull device) {
    NSString *log = [NSString stringWithFormat:@"ScanDevice
        Name :%@ UUID :%@ mac :%@", [device name], [device
        UUID], [device mac]];
    [weakself addLogWhitText:log];
} CancelScanBlock:^() {
    [weakself addLogWhitText:@"Cancel Scan"];
}];
```

## 6.2.5. monitor device connection status:

```
[self.sdk getConnectDeviceStateBlock:^(id _Nonnull device,
    CBCentralManager * _Nonnull central, CBPeripheral * _Nonnull
    peripheral, CBPeripheralState state, NSError * _Nonnull error)
{
    [weakself connectDeviceInfo:device PeripheralState:state];
} CancelAllDevicesConnectionBlock:^(
    [weakself addLogWhitText:@"Cancel All Devices Connection"];
}];
```

6.2.6. join Parallax device: when mobile phone Bluetooth After it is turned on, it will automatically scan the surrounding devices, find a matching device, and automatically connect to the device by default. After the device is connected, if no command is sent for more than 120 seconds, it will automatically disconnect.

```
[self.sdk device:@[self.oxygen]];
```

- 6.2.7. Monitoring command response status: Send commands according to usage needs, and obtain corresponding information through CMD.

```
[self.oxygen getReadDataModelBlock:^(NSInteger CMD, id _Nonnull model, NSError * _Nonnull error) {  
    [weakSelf log:weakSelf.oxygen CMD:CMD Model:model Error:error];  
}];
```

- 6.2.8. Monitor error response status: When a response is received, it will automatically disconnect.

```
[self.oxygen getValidationErrorBlock:^(NSString * _Nonnull item,  
    NSString * _Nonnull info, NSData * _Nonnull value) {  
  
}];
```

### 6.3. Device automatic scanning mechanism:

- 6.3.1. When CBManagerStatePoweredOn automatically opens scanning [Default: Open]
- 6.3.2. unified management mode (single device connection), use 3.5. Cancel the automatic scanning mechanism.

```
// stop Auto Scan
[self.sdk autoScan:NO];
```

- 6.3.3. Use independent management mode (multiple devices are connected at the same time), use 4.3. Cancel the automatic scanning mechanism.

```
[self.oxygen autoScan:NO];
```

### 6.4. Device automatic connection mechanism:

- 6.4.1. When a device is scanned, it will automatically connect to the device if it finds a matching device by default.
- 6.4.2. Use 4.4. Auto Connect to cancel the automatic connection mechanism.

```
[self.oxygen setAutoConnect:NO];
```

- 6.4.3. To monitor the scan device status, you can perform connection according to the response information of void(^scanDeviceBlock) (id device).

```
[self.sdk getScanDeviceBlock:^(id _Nonnull device) {
    NSString *log = [NSString stringWithFormat:@"ScanDevice
    Name : %@ UUID : %@ mac : %@", [device name], [device
    UUID], [device mac]];
    [weakself addLogWhitText:log];
    [weakself.oxygen connectDevice];
} CancelScanBlock:^() {
    [weakself addLogWhitText:@"Cancel Scan"];
}];
```

7. Dome Code description:

7.1. Dome Code navigation page:

- 7.1.1. A: Body Temperature SDK, function display of forehead temperature gun SDK.
- 7.1.2. B: Blood Pressure SDK, blood pressure machine SDK function display.
- 7.1.3. C: Oxygen SDK, function display of blood oxygen machine SDK.
- 7.1.4. D: Other SDK, go to the second navigation page.
- 7.1.5. E: Scan Device, Bluetooth SDK function display.
- 7.1.6. F: Cloud API, Cloud API SDK function display.
- 7.1.7. G: Send local logs to MicroLife.
- 7.1.8. H: One For All, automatic multi-device connection display, independent management mode (multi-device connection at the same time).
- 7.1.9. I: WatchBP O3 II SDK, WatchBP O3 II SDK function display.
- 7.1.10. J: WatchBP Office II SDK, WatchBP Office II SDK function display.
- 7.1.11. K: WatchBP HOME SDK, WatchBP Home SDK function display.
- 7.1.12. L: LTC, WatchBP M1 SDK function display.
- 7.1.13. M: Blood Sugar SDK, Blood Sugar SDK function display.
- 7.1.14. N: ESG SDK, ESG SDK function display.
- 7.1.15. O: Body Composition SDK, body fat machine SDK function display.





## 7.2. Oxygen SDK interface description:

7.2.1. A: Log display area.

7.2.2. B: Drawing area.

7.2.3. C: Function command area.

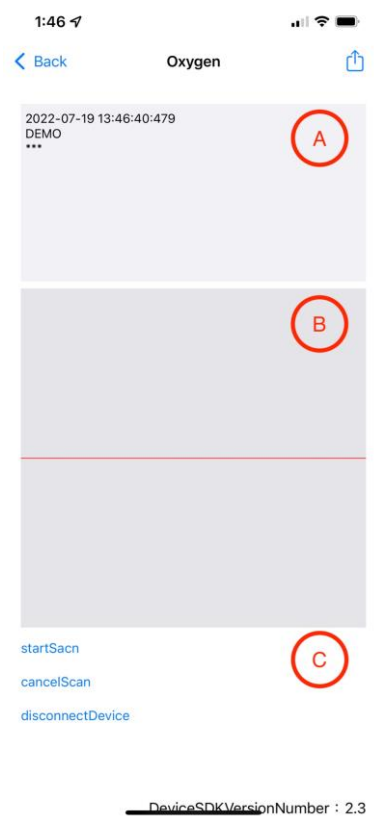
7.2.4. For related code, refer to OxygenViewController.

7.2.5. Instructions for use:

7.2.5.1. Entering the display page will automatically perform a device search.

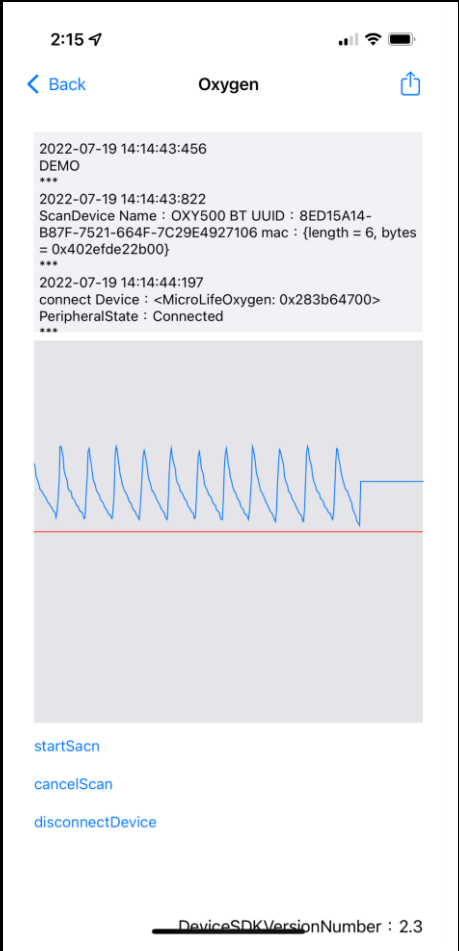
7.2.5.2. If a matching device is found, the device will be automatically connected.

7.2.5.3. After the device is connected, if no response is received for more than 120 seconds, it will automatically disconnect.

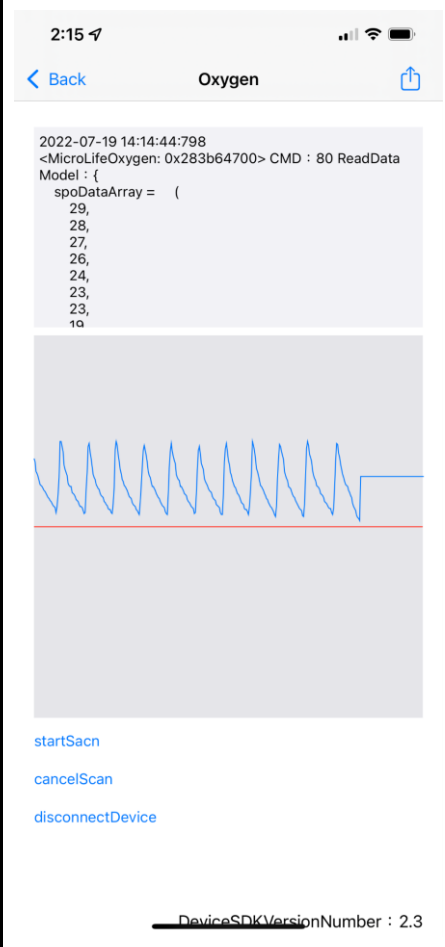


## 8. Operation Instructions

### 8.1. Search & Connect:

	<ol style="list-style-type: none"><li>1. After entering the display page, the SDK will automatically start searching and connecting to the matching device.</li><li>2. ScanDevice Name: OXY500 BT UUID: 8ED15A14-B87F-7521-664F-7C29E4927106 mac: {length = 6, bytes = 0x402efde22b00}</li><li>3. connect Device: &lt;MicroLifeOxygen: 0x283b47b80&gt; PeripheralState: Connected</li></ol>
--	---

## 8.2. Oxygen Volume Tracing Data:

	<pre>1.&lt; MicroLifeOxygen: 0x283b64700&gt; CMD: 80 ReadData Model: {   spoDataArray = (     29,     28,     27,     26,     24,     23,     23,     19,     19,     27   ); } Mac: 40:2E:FD:E2:2B:00 error : (null)</pre>
--	---

## 8.3. Oxygen Saturation And Pulse Rate Data:

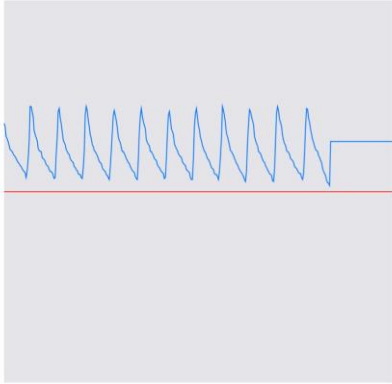
2:18

< Back Oxygen

----

2022-07-19 14:14:46:429  
<MicroLifeOxygen: 0x283b64700> CMD : 81 ReadData  
Model : {  
PI = 0;  
Plus = 102;  
SPO = 98;  
} Mac : 40:2E:FD:E2:2B:00 error : (null)  
\*\*\*

2022-07-19 14:14:46:565  
<MicroLifeOxygen: 0x283b64700> CMD : 80 ReadData  
Model : {



startSacn  
cancelScan  
disconnectDevice

DeviceSDKVersionNumber : 2.3

1. <MicroLifeOxygen:  
0x283b64700> CMD: 81  
ReadData Model: {  
PI = 0;  
Plus = 102;  
SPO = 98;  
} Mac: 40:2E:FD:E2:2B: 00 error:  
(null)

9. Update record:
  - 9.1. V2.x vs V1.x Difference description:
    - 9.1.1. All request responses are changed from delegate mode to block mode.
    - 9.1.2. Separate general-purpose tools and Perrier-specific tools.
    - 9.1.3. Data object that separates BPM and WatchBP.
    - 9.1.4. Added Bluetooth unified management mode (single device connection) and independent management mode (multiple devices connected at the same time).
  - 9.2. V2.0: Support device series BPM, BT, Oxygen
  - 9.3. V2.1: Support device series WatchBP home, WatchBP O3(2 schedule), WatchBP office
  - 9.4. V2.2: Support device series WatchBP O3(5 schedule)
  - 9.5. V2.3:
    - 9.5.1. Correction: - (void)startSacn; >> - (void)startScan;.
    - 9.5.2. Added: - (void)autoScan:(BOOL)autoScan;, When CBManagerStatePoweredOn automatically opens scanning [Default: Open]
    - 9.5.3. Fixed: Some commands that need to determine the device version are changed from - (void) to - (BOOL)

