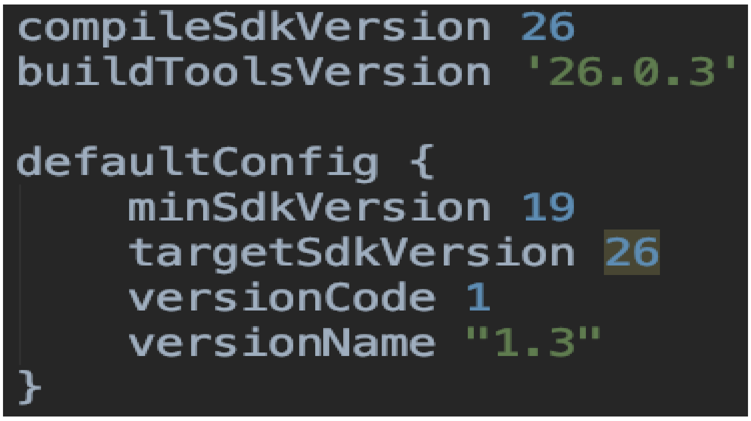
Demo APP for MicroLifeDeviceSDK (WatchBP O3) - Android

Table of Contents

Chapter 1 Development Environment

The supported SDK version is as follow:

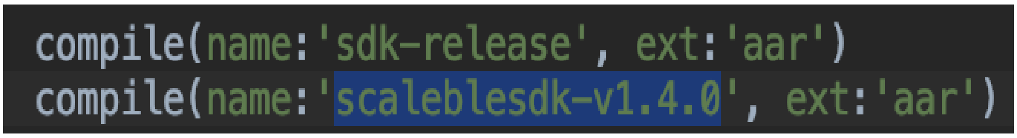


```
compileSdkVersion 26
buildToolsVersion '26.0.3'

defaultConfig {
    minSdkVersion 19
    targetSdkVersion 26
    versionCode 1
    versionName "1.3"
}
```

1.1. Add the library “sdk-release.arr” into the “libs” directory.

1.2. In the “build.gradle”, add the description as bellows:

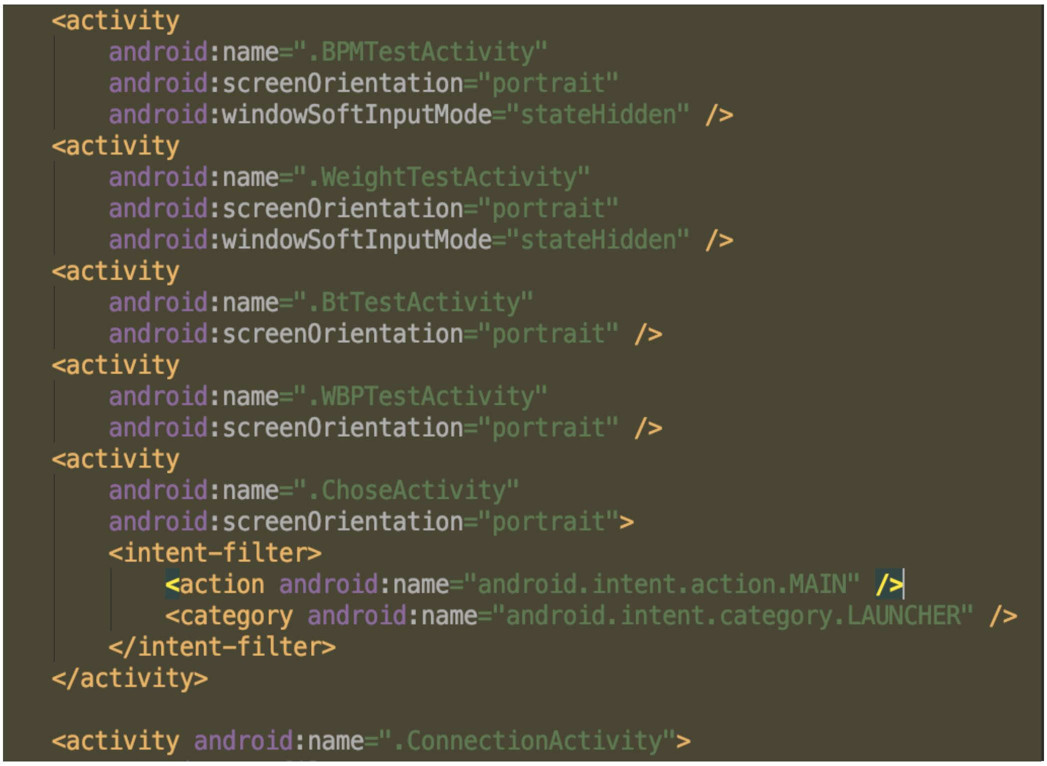


```
compile(name:'sdk-release', ext:'aar')
compile(name:'scaleblesdk-v1.4.0', ext:'aar')
```

Chapter 2 Entry Point and Bluetooth LE Protocol

The “ChoseActivity” is the entry point of the sample application.

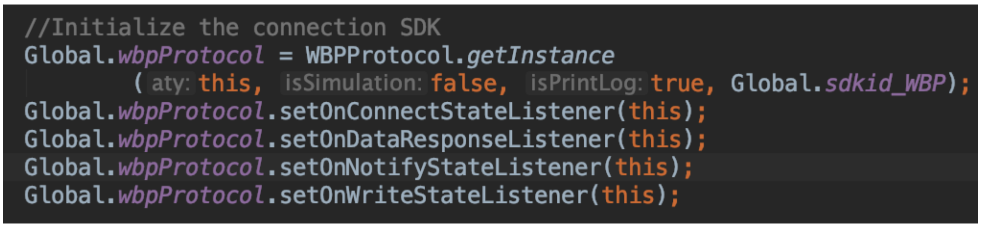
The “WBO3TestActivity” is dedicated to the device WatchBP O3 (Bluetooth LE).



```
<activity
    android:name=".BPMTestActivity"
    android:screenOrientation="portrait"
    android:windowSoftInputMode="stateHidden" />
<activity
    android:name=".WeightTestActivity"
    android:screenOrientation="portrait"
    android:windowSoftInputMode="stateHidden" />
<activity
    android:name=".BtTestActivity"
    android:screenOrientation="portrait" />
<activity
    android:name=".WBPTestActivity"
    android:screenOrientation="portrait" />
<activity
    android:name=".ChoseActivity"
    android:screenOrientation="portrait">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>

<activity android:name=".ConnectionActivity">
```

2.1 Initialize the instance “wbpProtocol”. This is to fulfill Bluetooth LE features and connection sequence.



```
//Initialize the connection SDK
Global.wbpProtocol = WBPProtocol.getInstance
        ( aty: this,  isSimulation: false,  isPrintLog: true, Global.sdkid_WBP);
Global.wbpProtocol.setOnConnectStateListener(this);
Global.wbpProtocol.setOnDataResponseListener(this);
Global.wbpProtocol.setOnNotifyStateListener(this);
Global.wbpProtocol.setOnWriteStateListener(this);
```

2.1.1 The “setOnConnectStateListener()” is to get the connection status of device.

2.1.2 The “setOnDataResponseListener()” is to get the response from device.

2.1.3 The “setOnNotifyStateListener()” is to get the data which is response from device.

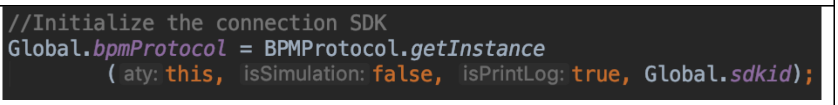
2.1.4 The “setOnWriteStateListener()” is to get the data which is sent to device.

2.2 The “isEnableBt()” or “ isSupportBluetooth() is to check if the smartphone’s Bluetooth is enabled or not. The “isSupportBluetooth()” will prompt a warning message to inform user to turn on Bluetooth if it is disabled.

Chapter 3 Bluetooth LE Protocol & APIs

3.1. Instance of Bluetooth LE Protocol：

3.1.1. Interface：

	public static * Protocol getInstance(Activity aty, boolean isSimulation, boolean isPrintLog, String sdkid)
Definition	Initialize Bluetooth LE Protocol for WatchBP Home A device
Parameter	Activity aty：name of activity or this boolean isSimulation： is simulator or device boolean isPrintLog：is printing log or not。 String sdkid：SDK ID of designated device
	

```
//Initialize the connection SDK
Global.bpmProtocol = BPMProtocol.getInstance
        ( aty: this,  isSimulation: false,  isPrintLog: true, Global.sdkid);
```

3.2. Connection State and Result：

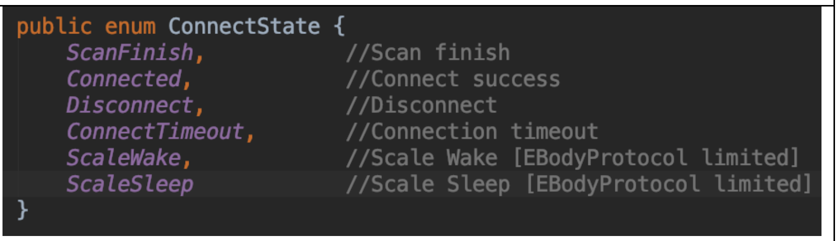
3.2.1. Interface：

	public void setOnConnectStateListener(OnConnectStateListener l)
Definition	The “setOnConnectStateListener()” is to get the connection status of device.

3.2.2. Delegate：

	void onBtStateChanged(boolean isEnable)
Definition	The “onBtStateChanged()” is to monitor the state of Enabled or Disabled.

	void onScanResult(String mac, String name, int rssi)
Definition	This is to get Bluetooth information of devices which discovered in the vicinity.
Parameter	macAddress: MAC of device name: device name RSSI: RSSI

	void onConnectionState(ConnectState state)
Definition	The "onConnectionState()" is to monitor the status of connection.
Parameter	

```
public enum ConnectState {
    ScanFinish,             //Scan finish
    Connected,              //Connect success
    Disconnect,             //Disconnect
    ConnectTimeout,         //Connection timeout
    ScaleWake,              //Scale Wake [EBodyProtocol limited]
    ScaleSleep              //Scale Sleep [EBodyProtocol limited]
}
```

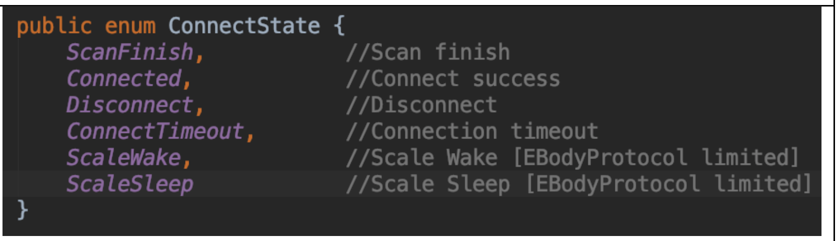
3.3. Device scanning or discovery：

3.3.1. Interface：

	public void startScan(int timeout)
Definition	The "startScan()" is for device scanning or discovery. The result will be shown with the "onScanResult".
Parameter	int timeout

	public void stopScan()
Definition	Terminate the scanning process.

3.3.2. Delegate：

	void onConnectionState(ConnectState state)
Definition	The "onConnectionState()" is to monitor the status of scanning.
Parameter	

```
public enum ConnectState {
    ScanFinish,             //Scan finish
    Connected,              //Connect success
    Disconnect,             //Disconnect
    ConnectTimeout,         //Connection timeout
    ScaleWake,              //Scale Wake [EBodyProtocol limited]
    ScaleSleep              //Scale Sleep [EBodyProtocol limited]
}
```

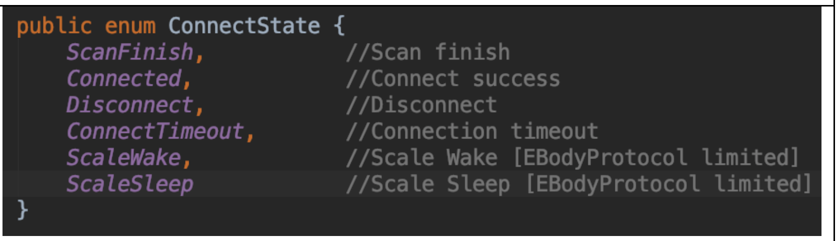
3.4. Connection：

3.4.1. Interface：

	public void connect(String macAddress)

Definition	Connect to device with MAC address.
Parameter	macAddress: MAC of device

3.4.2. Delegate：

	void onConnectionState(ConnectState state)
Definition	The “onConnectionState()” is to monitor the status of connection.
Parameter	

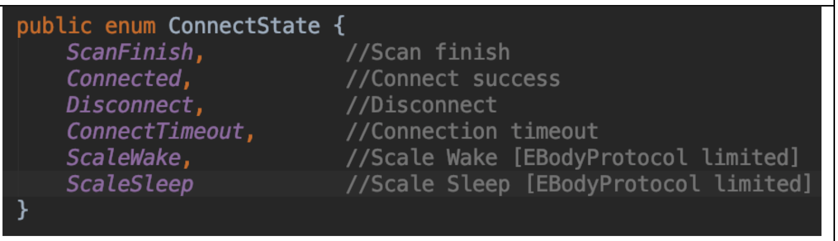
```
public enum ConnectState {
    ScanFinish,             //Scan finish
    Connected,              //Connect success
    Disconnect,             //Disconnect
    ConnectTimeout,         //Connection timeout
    ScaleWake,              //Scale Wake [EBodyProtocol limited]
    ScaleSleep              //Scale Sleep [EBodyProtocol limited]
}
```

3.5. Bonding：

3.5.1. Interface：

	public void bond(String macAddress)
Definition	Binding specified device by MAC
Parameter	macAddress: MAC of device

3.5.2. Delegate：

	void onConnectionState(ConnectState state)
Definition	The “onConnectionState()” is to monitor the status of connection.
Parameter	

```
public enum ConnectState {
    ScanFinish,             //Scan finish
    Connected,              //Connect success
    Disconnect,             //Disconnect
    ConnectTimeout,         //Connection timeout
    ScaleWake,              //Scale Wake [EBodyProtocol limited]
    ScaleSleep              //Scale Sleep [EBodyProtocol limited]
}
```

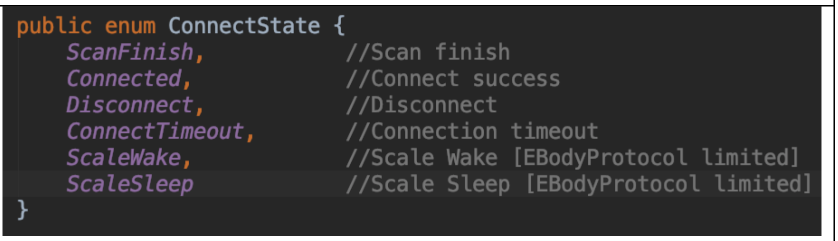
3.6. Disconnection：

3.6.1. Interface：

	public void disconnect()

Definition	Disconnect device.

3.6.2. Delegate :

	void onConnectionState(ConnectState state)
Definition	The “onConnectionState()” is to monitor the status of disconnection.
Parameter	

Parameter:

```
public enum ConnectState {
    ScanFinish,           //Scan finish
    Connected,            //Connect success
    Disconnect,           //Disconnect
    ConnectTimeout,       //Connection timeout
    ScaleWake,            //Scale Wake [EBodyProtocol limited]
    ScaleSleep            //Scale Sleep [EBodyProtocol limited]
}
```

Chapter 4 WatchBP O3 APIs

4.1. Read all history or current data from BPM：

4.1.1. Interface：

	public void readAllHistorys()
Definition	Read all history or current data from BPM

4.1.2. Delegate：

	void onResponseReadAllHistorys(DRecord dRecord)
Parameter	dRecord： The data is from the mode of History Measurement.

4.2. Read central BP memory data by index from BPM：

4.2.1. Interface：

	public void readCBPData(int index, CBPdataAndCalCBP.Dformat dformat)
Definition	Read central BP memory data by index from BPM
Parameter	index：Memory index from CBP dformat： Data format

4.2.2. Delegate：

	void onResponseReadCBPData(CBPdataAndCalCBP cRecord,boolean isNullData)
Parameter	cRecord：CBP data & CalCBP data isNullData：True or False

4.3. Clear all history data of the BPM：

4.3.1. Interface：

	public void clearAllHistorys()
Definition	Clear all history data of the BPM

4.3.2. Delegate：

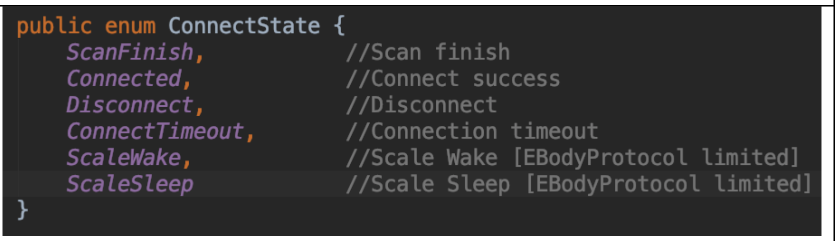
	void onResponseClearHistorys(boolean isSuccess)
Parameter	isSuccess：True or False

4.4. Disconnect the Bluetooth with BPM：

4.4.1. Interface：

	public void disconnectWBO3()
Definition	Disconnect device.

4.4.2. Delegate：

	void onConnectionState(ConnectState state)
Definition	The “onConnectionState()” is to monitor the status of disconnection.
Parameter	

```
public enum ConnectState {
    ScanFinish,           //Scan finish
    Connected,            //Connect success
    Disconnect,           //Disconnect
    ConnectTimeout,       //Connection timeout
    ScaleWake,            //Scale Wake [EBodyProtocol limited]
    ScaleSleep            //Scale Sleep [EBodyProtocol limited]
}
```

4.5. Read user ID and version data from BPM：

4.5.1. Interface：

	public void readUserAndVersionData()
Definition	Read user ID and version data from BPM

4.5.2. Delegate：

	void onResponseReadUserAndVersionData(User user, VersionData verData)
Parameter	user：user ID verData：version data

4.6. Write a new user ID to BPM：

4.6.1. Interface：

	public void writeUserID(String ID)
Definition	Write a new user ID to BPM
Parameter	ID：user ID。

4.6.2. Delegate：

	void onResponseWriteUserID(boolean isSuccess)
Parameter	isSuccess：True or False

4.7. Read ABPM setting values from BPM：

4.7.1. Interface：

	public void readSettingValues()
Definition	Read ABPM setting values from BPM

4.7.2. Delegate：

	void onResponseReadSettingValues(SettingValues settingValues)
Parameter	settingValues： ABPM setting values

4.8. Write ABPM setting values to BPM ：

4.8.1. Interface：

	public void writeSettingValues(SettingValues settingValues)
Definition	Write ABPM setting values to BPM
Parameter	settingValues：ABPM setting values

4.8.2. Delegate：

	void onResponseWriteSettingValues(boolean isSuccess)
Parameter	isSuccess：True or False

4.9. Read device ID and info from BPM：

4.9.1. Interface：

	public void readDeviceIDAndInfo()

Definition	Read device ID and info from BPM

4.9.2. Delegate：

	void onResponseReadDeviceInfo(DeviceInfo deviceInfo)
Parameter	deviceInfo：device ID and info

4.10. Read device Time from BPM：

4.10.1. Interface：

	public void readDeviceTime()
Definition	Read device Time from BPM

4.10.2. Delegate：

	void onResponseReadDeviceTime(DeviceInfo deviceInfo)
Parameter	deviceInfo：device Time

4.11. Write device Time to BPM：

4.11.1. Interface：

	public void writeDeviceTime()
Definition	Write device Time to BPM

4.11.2. Delegate：

	void onResponseWriteDeviceTime(boolean isSuccess)
Parameter	isSuccess：True or False

4.12. Read BPM function setting value from BPM：

4.12.1. Interface：

	public void readFunctionSettingValue()
Definition	Read BPM function setting value from BPM

4.12.2. Delegate：

	void

	onResponseReadFunctionSettingValues(FunctionSettingValues functionSettingValues)
Parameter	functionSettingValues：BPM function setting value

4.13. Read BT module name from BPM：

4.13.1. Interface：

	public void readBTModuleName()
Definition	Read BT module name from BPM

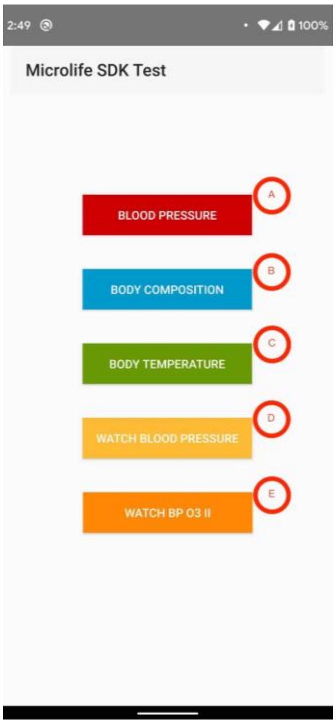
4.13.2. Delegate：

	void onResponseReadBTModuleName(String BTModuleName)
Parameter	BTModuleName：BT Module Name

Chapter 5 User Interface of Demo App

5.1 Getting Started：

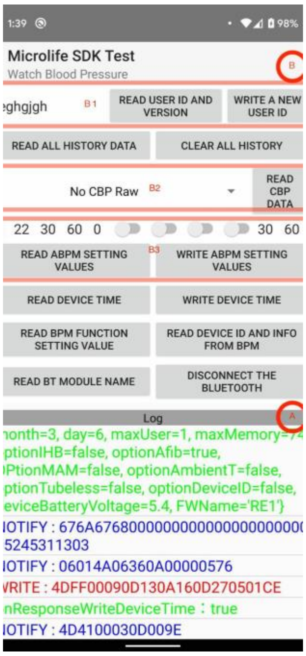
Start the app and then select the button “WATCH BP O3 II” / “E” to communicate with the designate device WatchBP O3 II.



5.2 Operation Sequence：

The scanning (discovery) is automatically run to discover devices in the vicinity. If a device is bonded, it will be connected accordingly.

5.3 GUI Layout :



```
ıonth=3, day=6, maxUser=1, maxMemory=
ptionIHB=false, optionAfib=true,
PtionMAM=false, optionAmbientT=false,
ptionTubeless=false, optionDeviceID=false,
eviceBatteryVoltage=5.4, FWName='RE1'}
IOTIFY : 676A676800000000000000000000
5245311303
IOTIFY : 06014A06360A00000576
VRITE : 4DFF00090D130A160D270501CE
nResponseWriteDeviceTime：true
IOTIFY : 4D4100030D009E
```

5.3.1 Region A：The log window is used to display information about communication handshake between App and device.

5.3.2 Region B：This part is to communicate with the device WatchBP O3 II by different functions / commands such as data transferring, synchronization and so on.

B1：Read/ Write User ID。

B2：Read CBP data with index. Firstly, the function “READ ALL HISTORY DATA” shall be run before performing this function.

B3：Read/ Write ABPM setting values as follows:

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 ABPM setting values are the following:

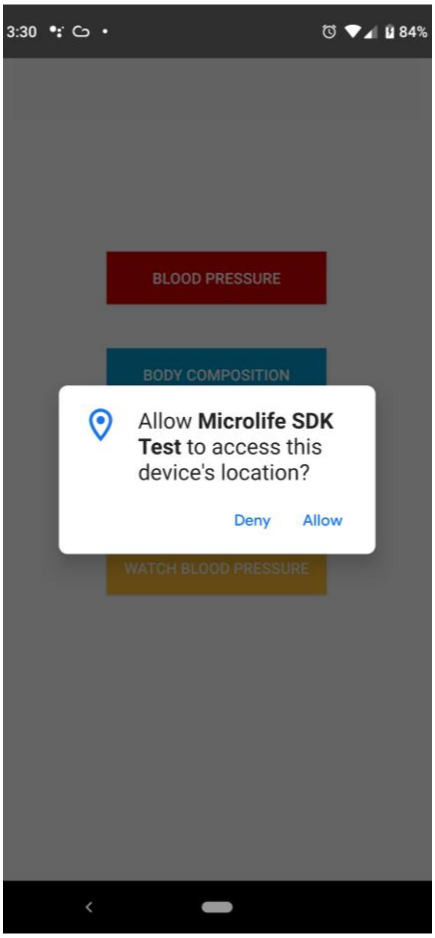
ABPMStart=6,
ABPMEnd=23,
ABPMInt_first=10,
ABPMInt_second=20,
HI_infPressure=200,
CBP_zone2_meas_off=true,
CBP_zone1_meas_off=false,
SW_SEL_silent=false,

```
SW_checkhide=true,
CBPInt_first=30,
CBPInt_second=40
```

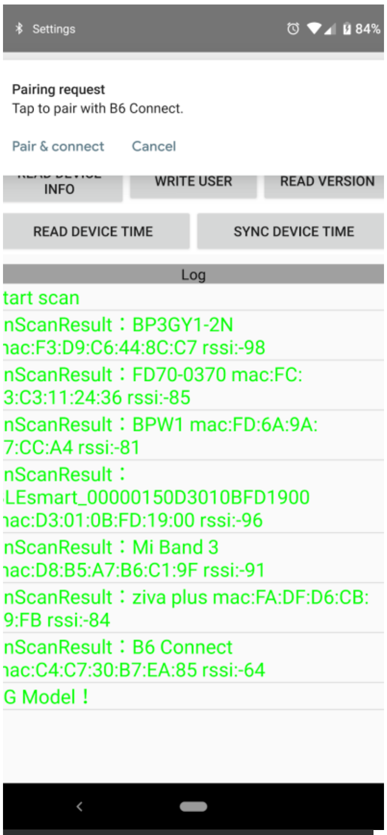
5.4 Refer to “WBO3TestActivity” from the demo application (sample code) to get more detailed.

Chapter 6　Functionality of Demo App

6.1.　Bluetooth authorization：

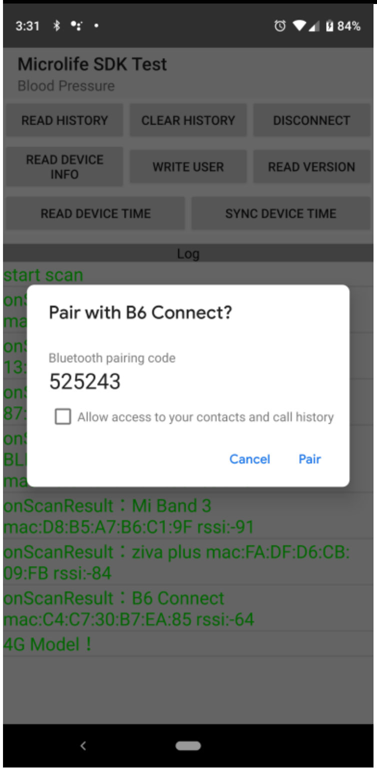
 Allow **Microlife SDK Test** to access this device's location? 3:30 84% BLOOD PRESSURE BODY COMPOSITION WATCH BLOOD PRESSURE Deny Allow	1. Request for Bluetooth permission.

6.2 Pairing / Bonding：



Pairing request
Tap to pair with B6 Connect.

```
tart scan
nScanResult：BP3GY1-2N
ac:F3:D9:C6:44:8C:C7 rssi:-98
nScanResult：FD70-0370 mac:FC:
3:C3:11:24:36 rssi:-85
nScanResult：BPW1 mac:FD:6A:9A:
7:CC:A4 rssi:-81
nScanResult：
LEsmart_00000150D3010BFD1900
ac:D3:01:0B:FD:19:00 rssi:-96
nScanResult：Mi Band 3
ac:D8:B5:A7:B6:C1:9F rssi:-91
nScanResult：ziva plus mac:FA:DF:D6:CB:
9:FB rssi:-84
nScanResult：B6 Connect
ac:C4:C7:30:B7:EA:85 rssi:-64
G Model！
```



Pair with B6 Connect?

Bluetooth pairing code
525243

☐ Allow access to your contacts and call history

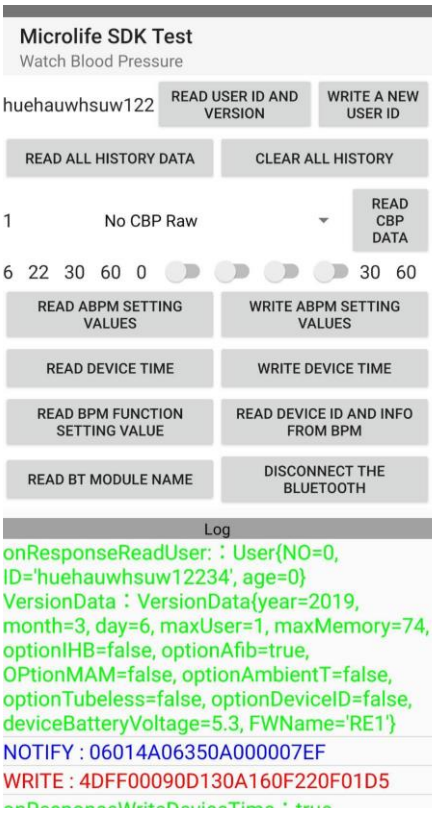
```
start scan
on
ma
on
13:
on
87:
on
BL
ma
onScanResult：Mi Band 3
mac:D8:B5:A7:B6:C1:9F rssi:-91
onScanResult：ziva plus mac:FA:DF:D6:CB:
09:FB rssi:-84
onScanResult：B6 Connect
mac:C4:C7:30:B7:EA:85 rssi:-64
4G Model！
```

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

2. Once the procedure is done, choose any function/ command to do communication with WatchBP O3 II device.

3. The green part is from “onScanResult”.

6.3 Command: Write a new user ID to BPM



```
onResponseReadUser: ：User{NO=0,
ID='huehauwhsuw12234', age=0}
VersionData ：VersionData{year=2019,
month=3, day=6, maxUser=1, maxMemory=74,
optionIHB=false, optionAfib=true,
OPtionMAM=false, optionAmbientT=false,
optionTubeless=false, optionDeviceID=false,
deviceBatteryVoltage=5.3, FWName='RE1'}
NOTIFY : 06014A06350A000007EF
WRITE : 4DFF00090D130A160F220F01D5
```

1. The command “Write a new user ID” is to write a new user ID to device.

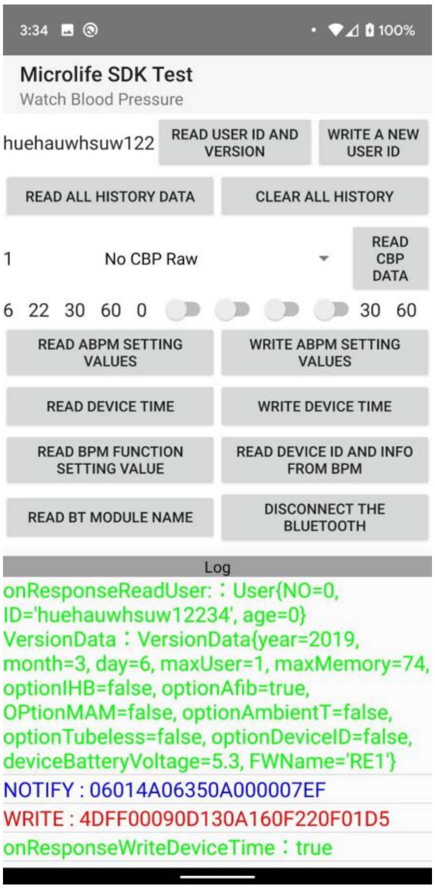
2.WRITE :writeUserID：
huehauwhsuw12234
The ID “huehauwhsuw12234” is made up of ASCII code.

3.onResponseWriteUserID：
true：
This means that the writing/ sending procedure is successful.

4. It can be verifued by utilizing the command “Read user ID and version data”.

5. The red part is the command and communication protocol that is sent to device.
The blue part is the raw data from device via Bluetooth.
The green part is from “onResponseWriteUserID” which is decoded from the raw data.

6.4 Command: Read user ID and version data from BPM

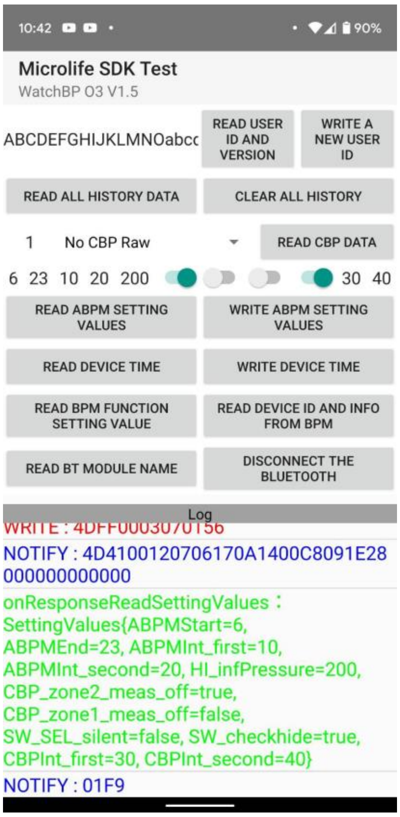


```
onResponseReadUser:：User{NO=0,
ID='huehauwhsuw12234', age=0}
VersionData：VersionData{year=2019,
month=3, day=6, maxUser=1, maxMemory=74,
optionIHB=false, optionAfib=true,
OPtionMAM=false, optionAmbientT=false,
optionTubeless=false, optionDeviceID=false,
deviceBatteryVoltage=5.3, FWName='RE1'}
NOTIFY : 06014A06350A000007EF
WRITE : 4DFF00090D130A160F220F01D5
onResponseWriteDeviceTime：true
```

1. The command “Read user ID and version data” is to get user ID and device information as below.

2. onResponseReadUser: : User{NO=0, ID=huehauwhsuw12234, age=0} VersionData : VersionData{year=2019, month=3, day=6, maxUser=1, maxMemory=74, optionIHB=false, optionAfib=true, OPtionMAM=false, optionAmbientT=false, optionTubeless=false, optionDeviceID=false, deviceBatteryVoltage=5.3, FWName='RE1'’}

6.5 Command: Read ABPM setting values from BPM

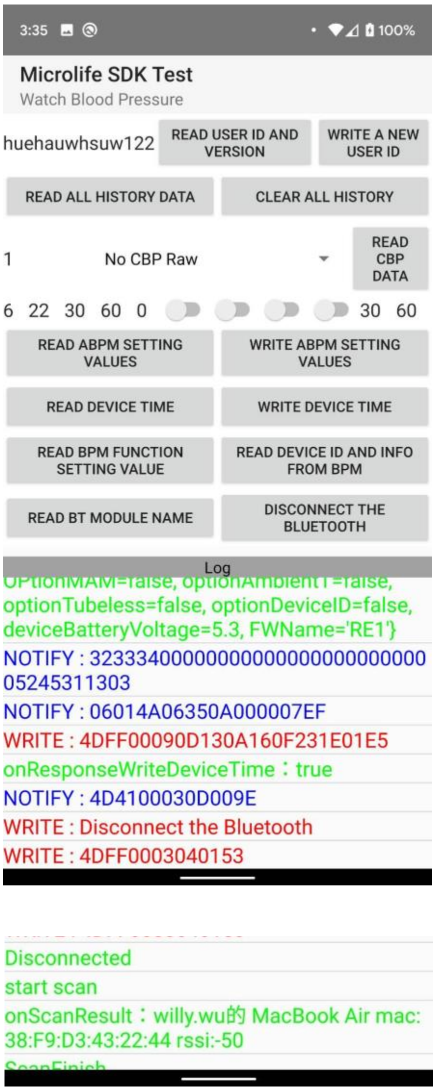


```
WRITE : 4DFF0003070156
NOTIFY : 4D4100120706170A1400C8091E28
000000000000
onResponseReadSettingValues：
SettingValues{ABPMStart=6,
ABPMEnd=23, ABPMInt_first=10,
ABPMInt_second=20, HI_infPressure=200,
CBP_zone2_meas_off=true,
CBP_zone1_meas_off=false,
SW_SEL_silent=false, SW_checkhide=true,
CBPInt_first=30, CBPInt_second=40}
NOTIFY : 01F9
```

1. This is to perform the function Read ABPM setting values from BPM.

2.onResponseReadSettingValues : SettingValue{
ABPMStart=6,
ABPMEnd=23,
ABPMInt_first=10,
ABPMInt_second=20,
HI_infPressure=200,
CBP_zone2_meas_off=true,
CBP_zone1_meas_off=false,
SW_SEL_silent=false,
SW_checkhide=true,
CBPInt_first=30,
CBPInt_second=40
}

6.6 Command: Disconnect the Bluetooth



huehauwhsuw122

1 No CBP Raw

6 22 30 60 0 30 60

```
optionMAM=false, optionAmbientT=false,
optionTubeless=false, optionDeviceID=false,
deviceBatteryVoltage=5.3, FWName='RE1'}
NOTIFY : 32333400000000000000000000000
05245311303
NOTIFY : 06014A06350A000007EF
WRITE : 4DFF00090D130A160F231E01E5
onResponseWriteDeviceTime：true
NOTIFY : 4D4100030D009E
WRITE : Disconnect the Bluetooth
WRITE : 4DFF0003040153
```

```
Disconnected
start scan
onScanResult：willy.wu的 MacBook Air mac:
38:F9:D3:43:22:44 rssi:-50
ScanFinish
```

1. Disconnect the Bluetooth

2. The command is “4DFF0003040153” and it is sent to device.

3. The result is able to observe by the “onConnectionState(ConnectState state)” and its status can be found by the “ConnectState = Disconnect”.

4. The “Disconnected” will be displayed on GUI layout. And then, the scanning (discovery) is automatically run to discover devices in the vicinity.