

# Demo APP for MicroLifeDeviceSDK (WatchBP Vascular) - Android

## Table of Contents

<b>Chapter 1</b>	Development Environment
<b>Chapter 2</b>	Entry Point and Bluetooth LE Protocol
<b>Chapter 3</b>	Bluetooth LE Protocol & APIs
<b>Chapter 4</b>	WatchBP Vascular APIs
<b>Chapter 5</b>	User Interface of Demo App
<b>Chapter 6</b>	Functionality of Demo App

## Chapter 1 Development Environment

The supported SDK version is as follow:

```
compileSdk 36
buildToolsVersion "36.0.0"

defaultConfig { DefaultConfig it ->
    applicationId "com.ideabus.sdk_test"
    minSdkVersion 29
    targetSdk 36
    versionCode 1
    versionName "2.1.8"
}
```

- 1.1. Add the library “\*.arr” or \*.jar into the “libs” directory.
- 1.2. In the “app\build.gradle” , add the description as bellows:

```
implementation fileTree(include: ["*.jar", "*.aar"], dir: "libs")
```

## Chapter 2 Entry Point and Bluetooth LE Protocol

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

The “.WBVTestActivity” is dedicated to the device WatchBP Vascular (Bluetooth LE).

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    <application
        android:screenOrientation="portrait"
        tools:ignore="LockedOrientationActivity" />
        <activity
            android:name=".SP02TestActivity"
            android:screenOrientation="portrait"
            tools:ignore="LockedOrientationActivity" />
        <activity
            android:name=".WBPTTestActivity"
            android:screenOrientation="portrait"
            tools:ignore="LockedOrientationActivity" />
        <activity
            android:name=".WB03TestActivity"
            android:screenOrientation="portrait"
            tools:ignore="LockedOrientationActivity" />
        <activity
            android:name=".WB0TestActivity"
            android:screenOrientation="portrait"
            tools:ignore="DiscouragedApi, LockedOrientationActivity" />

        <activity
            android:name=".WBVTestActivity"
            android:screenOrientation="portrait"
            tools:ignore="DiscouragedApi,LockedOrientationActivity" />

        <activity
            android:name=".ChoseActivity"
            android:screenOrientation="portrait"
            android:exported="true"
            tools:ignore="DiscouragedApi, LockedOrientationActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
```

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

```
//Initialize the connection SDK
Global.wbvProtocol = WBVProtocol.getInstance(activity this,
false, true, Global.sdkid_WBP);
Global.wbvProtocol.setOnConnectStateListener(this);
Global.wbvProtocol.setOnDataResponseListener(this);
Global.wbvProtocol.setOnNotifyStateListener(this);
Global.wbvProtocol.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 “isEnabledBt()” 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 Vascular 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
	<pre>//Initialize the connection SDK Global.wbvProtocol = WBVProtocol.getInstance(this, false, true, Global.sdkid_WBP);</pre>

### 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 isEnabled)
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	<pre>public enum ConnectState {     ScanFinish,           //Scan finish     Connected,            //Connect success     Disconnect,           //Disconnect     ConnectTimeout        //Connection timeout }</pre>	
-----------	---	--

### 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	<pre>public enum ConnectState {     ScanFinish,           //Scan finish     Connected,            //Connect success     Disconnect,           //Disconnect     ConnectTimeout        //Connection timeout }</pre>

### 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	<pre>public enum ConnectState {     ScanFinish,           //Scan finish     Connected,            //Connect success     Disconnect,           //Disconnect     ConnectTimeout        //Connection timeout }</pre>

## 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	<pre>public enum ConnectState {     ScanFinish,           //Scan finish     Connected,            //Connect success     Disconnect,           //Disconnect     ConnectTimeout        //Connection timeout }</pre>

## 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	<pre>public enum ConnectState {     ScanFinish,           //Scan finish     Connected,            //Connect success     Disconnect,           //Disconnect     ConnectTimeout        //Connection timeout }</pre>

## Chapter 4 WatchBP Vascular 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.DformatWBV dformat)
Definition	Read central BP memory data by index from BPM
Parameter	index : Memory index from CBP/PWV or ABI dformat : Data format NoCBPRaw : No CBP raw data LowCBPRaw : low resolution CBP/PWV data (sampling rate =16Hz) FullCBPRaw : full CBP raw data (sampling rate=256Hz) FullPWVRaw : full PWV/ABI raw data (sampling rate=1024Hz)

#### 4.2.2. Delegate :

	void onResponseReadCBPData(CBPdataAndCalCBP cRecord, boolean isNullData)
Parameter	cRecord : CBP data & CalCBP data isNullData : True or False
	void onResponseReadPWVData(PWVAndABIdata cRecord, boolean isNullData)
Parameter	cRecord : PWV or ABI 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 disconnectWBO()
Definition	Disconnect device.

#### 4.4.2. Delegate :

	void onConnectionState(ConnectState state)
Definition	The “onConnectionState()” is to monitor the status of disconnection.
Parameter	<pre>public enum ConnectState {     ScanFinish,           //Scan finish     Connected,            //Connect success     Disconnect,           //Disconnect     ConnectTimeout        //Connection timeout }</pre>

### 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, VersionDataWBV 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 BPM setting values from BPM :

#### 4.7.1. Interface :

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

#### 4.7.2. Delegate :

	void onResponseReadSettingValues(SettingsValues settingsValues)
Parameter	settingsValues : BPM setting values

### 4.8. Write BPM setting values to BPM :

#### 4.8.1. Interface :

	public void writeSettingValues(SettingsValues settingsValues)
Definition	Write BPM setting values to BPM Note: Below are illegal settings (1)AutoMeasureNumber=1 or 2 and SW_AVG_no_include_first=true (2)AutoMeasureNumber=1 and SW_AFib =true (3)Height < 120, Height> 210
Parameter	settingsValues : BPM 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 onResponseReadDeviceInfoWBV(DeviceInfoWBV 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

#### 4.14. Start remote measurement :

##### 4.14.1. Interface :

	public void startRemoteMeasurement( CBPdataAndCalCBP.DformatWBV dformat)
Definition	Start remote measurement
Parameter	dformat : Data format NoCBPRaw : No CBP raw data LowCBPRaw : low resolution CBP data (sampling rate =16Hz) FullCBPRaw : full CBP raw data (sampling rate=256Hz) FullPWVRaw : full PWV/ABI raw data (sampling rate=1024Hz)

##### 4.14.2. Delegate :

	void onResponseStartRemoteMeasurement( CBPdataAndCalCBP.DformatWBV dformat)
Definition	Start remote measurement
Parameter	dformat : Data format NoCBPRaw : No CBP raw data LowCBPRaw : low resolution CBP data (sampling rate =16Hz) FullCBPRaw : full CBP raw data (sampling rate=256Hz) FullPWVRaw : full PWV/ABI raw data (sampling rate=1024Hz)

	void onResponseRemoteMeasurementStatusEvery5seconds(STATUS status, int measurementNumber,int totalMeasurementNumber,int countdown,int totalMeasuretime)
Definition	Send remote measurement status every 5 seconds
Parameter	status : MeasurementWait : device wait countdown for next measurement. MeasurementStart : device is start BP measurement, MeasurementStop : manual press I/O to stop

	measurement measurementNumber : Send current measurement number in auto mode. totalMeasurementNumber : Send total measurement number in auto mode. countdown : Send current countdown time in auto mode. totalMeasuretime : Send total measurement time (seconds) in auto mode. Total measurement are count between 1st measurement to last measurement. (exclude rest time)
	void onResponseMeasurementResultsForEachMeasurement(CurrentAndMData dRecord, int historyMeasuremeNumber,int currentMeasurementTimes, boolean isAverage)
Definition	Send measurement results for each measurement
Parameter	dRecord : CurrentAndMData historyMeasuremeNumber : The history measurement times store in memory. currentMeasurementTimes : Send current measurement times to APP. isAverage : Send Average calculation when measurement to APP.

#### 4.15. Stop remote measurement :

##### 4.15.1 Interface

	public void stopRemoteMeasurement()
Definition	Stop remote measurement

#### 4.16. Read serial number from BPM :

##### 4.16.1 Interface :

	public void readSerialNumber()
Definition	Read BPM serial number from BPM

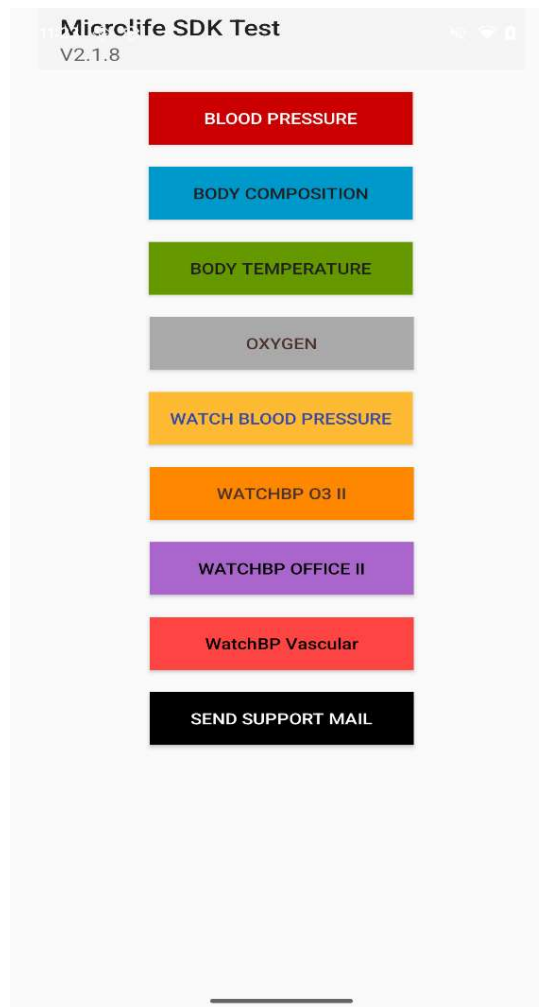
##### 4.16.2 Delegate :

	void onResponseReadSerialNumber(DeviceInfo deviceInfo)
Parameter	deviceInfo : serial number

## 4 User Interface of Demo App

### 5.1 Getting Started :

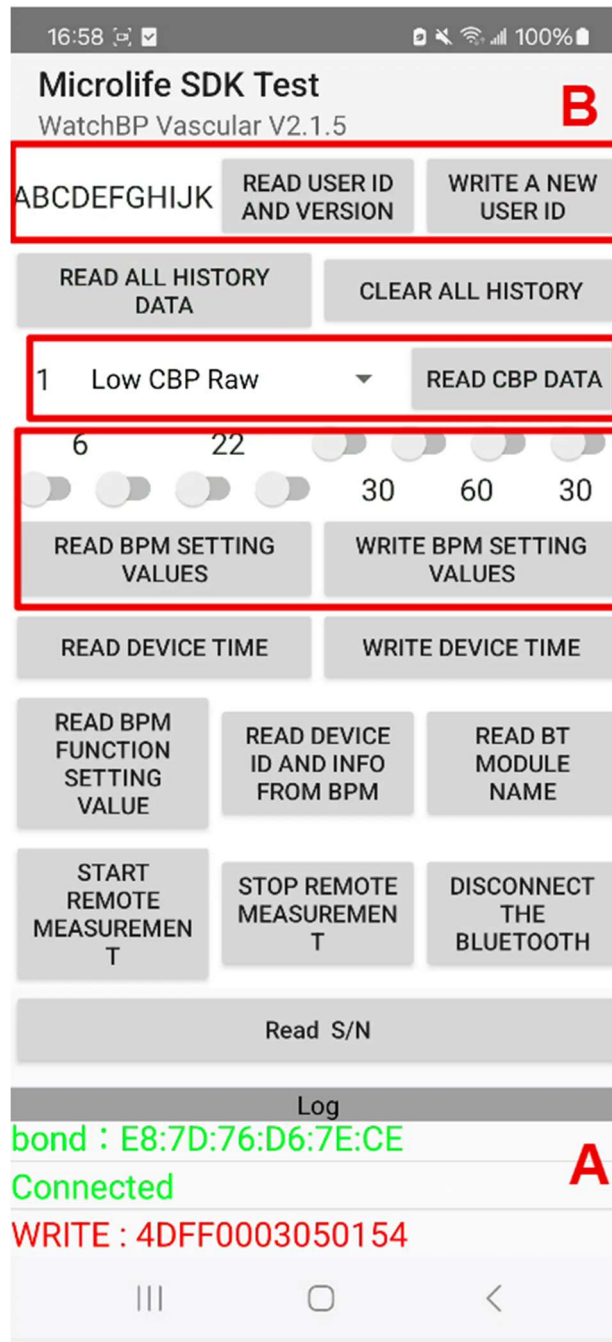
Start the app and then select the button “WATCH BP Vascular” to communicate with the designate device WatchBP Vascular.



### 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 :



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 Vascular by different functions / commands such as data transferring, synchronization and so on.

B1 : Read/ Write User ID.

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

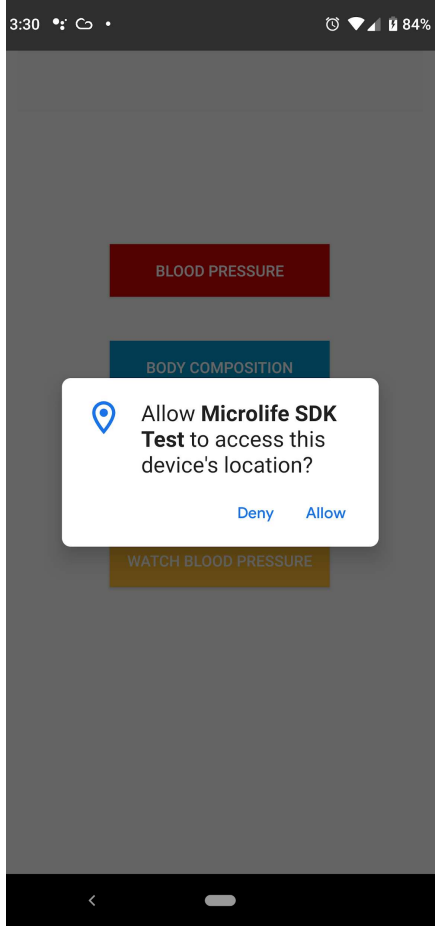
B3 : Read/ Write BPM setting values as follows:

1. Height: range 120 ~ 210(cm)
2. Highest inflation pressure of Auto mode :  
Valid parameter: 0(not setting), 160, 180, 200, 220, 240.The device inflates the cuff using fuzzy logic to proper cuff pressure; the Highest Inflation Pressure is considered as a safeguard pressure.
3. SW\_AUTO\_hide : Set Show readings during rest time in auto mode. true:hide/false:show.
4. SW\_AUS\_Hide : Set Show cuff pressure during deflation in AUS mode.  
true:hide/false:show.
5. SW\_AVG\_no\_include\_first : Set Average is include first memory data. true:is/false:is not.
6. SW\_CBP : Set CBP measurement  
true:enabled/false:disabled.
7. SW\_AFib : Set AFib measurement  
true:enabled/false:disabled.
8. SW\_AMP : Set 12/24-hour clock true:12-hour/false:24-hour.
9. SW\_Kpa : Set Pressure unit:  
true:Kpa/false:mmHg.
10. RestTime : Rest time of auto mode.Start countdown base on rest time before 1st measurement in auto mode.
11. IntervalTime : Interval time of auto mode.Start countdown base on interval time before 2nd~6th measurement in auto mode.
12. AutoMeasureNumber: It' s number of measurements in auto mode.
13. Note:Below are illegal settings
  - (1)AutoMeasureNumber=1 or 2 and SW\_AVG\_no\_include\_first=true
  - (2)AutoMeasureNumber=1 and SW\_AFib=true
  - (3) )Height < 120, Height> 210

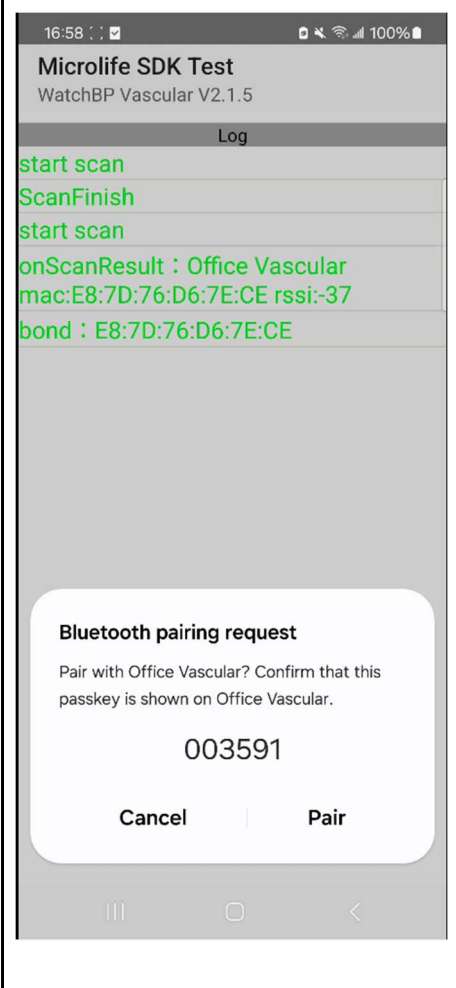
5.4 Refer to “WBVTestActivity” from the demo application (sample code) to get more detailed.

## Functionality of Demo App

### 6.1. Bluetooth authorization :

 <p>The screenshot shows the Microlife SDK Test app interface. At the top, there's a status bar with the time 3:30, signal strength, Wi-Fi, and battery at 84%. The app has a dark grey background with three buttons: 'BLOOD PRESSURE' (dark red), 'BODY COMPOSITION' (teal), and 'WATCH BLOOD PRESSURE' (olive green). A white location permission dialog is centered on the screen, asking 'Allow Microlife SDK Test to access this device's location?' with 'Deny' and 'Allow' options. The bottom of the screen shows a black navigation bar with a back arrow and a home indicator.</p>	<p>1. Request for Bluetooth permission.</p>
---	---

## 6.2 Pairing / Bonding :

	<ol style="list-style-type: none"><li>1. There is a message to confirm the pairing bonding procedure between device and cellphone if they haven't bonded yet.</li><li>2. Once the procedure is done, choose any function/ command to do communication with WatchBP Vascular device.</li><li>3. The <b>green</b> part is from "onScanResult" .</li></ol>
--	---



## 6.4 Command: Read user ID and version data from BPM

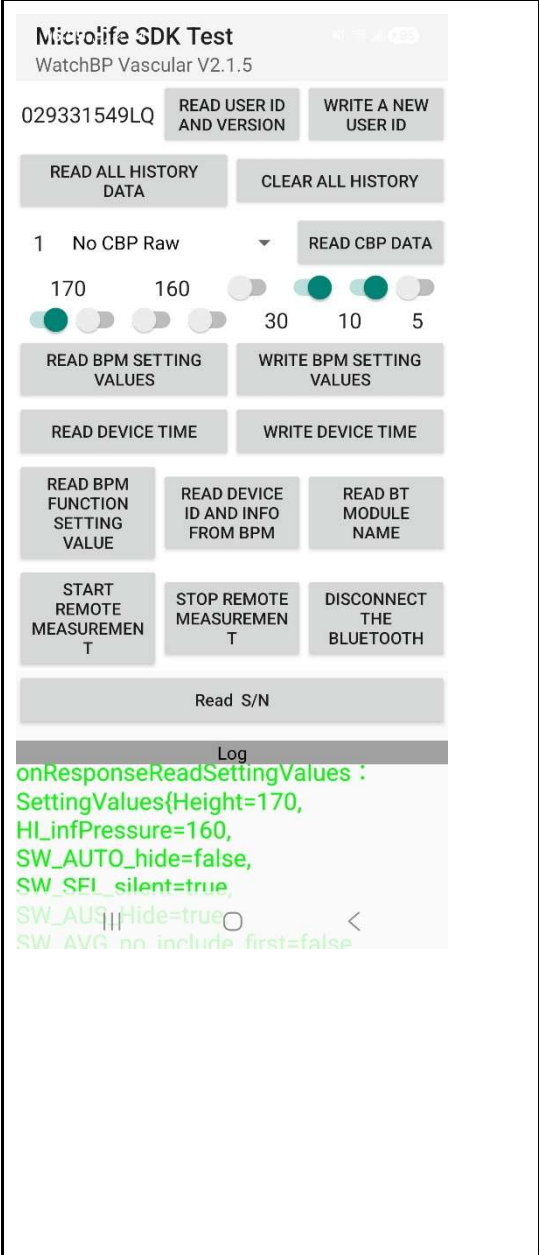
<p><b>MicroLife SDK Test</b> WatchBP Vascular V2.1.5</p> <p>029331549LQ <b>READ USER ID AND VERSION</b> <b>WRITE A NEW USER ID</b></p> <p><b>READ ALL HISTORY DATA</b> <b>CLEAR ALL HISTORY</b></p> <p>1 No CBP Raw <b>READ CBP DATA</b></p> <p>170 160 30 10 5</p> <p><b>READ BPM SETTING VALUES</b> <b>WRITE BPM SETTING VALUES</b></p> <p><b>READ DEVICE TIME</b> <b>WRITE DEVICE TIME</b></p> <p><b>READ BPM FUNCTION SETTING VALUE</b> <b>READ DEVICE ID AND INFO FROM BPM</b> <b>READ BT MODULE NAME</b></p> <p><b>START REMOTE MEASUREMENT</b> <b>STOP REMOTE MEASUREMENT</b> <b>DISCONNECT THE BLUETOOTH</b></p> <p><b>Read S/N</b></p> <p><b>Log</b></p>	<p>1. The command “Read user ID and version data” is to get user ID and device information as below.</p> <p>2. onResponseReadUser: :        User{NO=1, ID='029331549LQ', age=0}        VersionData :        VersionData{year=2025, month=5, day=8, maxUser=0, maxMemory=6, optionIHB= false, optionAfib= true, optionCBP=true, optionABI=true, optionPWV=true, optionPAD=false, deviceBatteryVoltage=6.5, FWName='RP3', ProtocolVersion= 1.0.0.3}</p>
<pre>onResponseReadUser: : User{NO=0, ID='029331549LQ', age=0} VersionData : VersionDataWBV{year=2025, month=5, day=8, maxUser=0, maxMemory=6, optionDeviceID=false}</pre>	

## 6.5 Command: Read All History Data from BPM

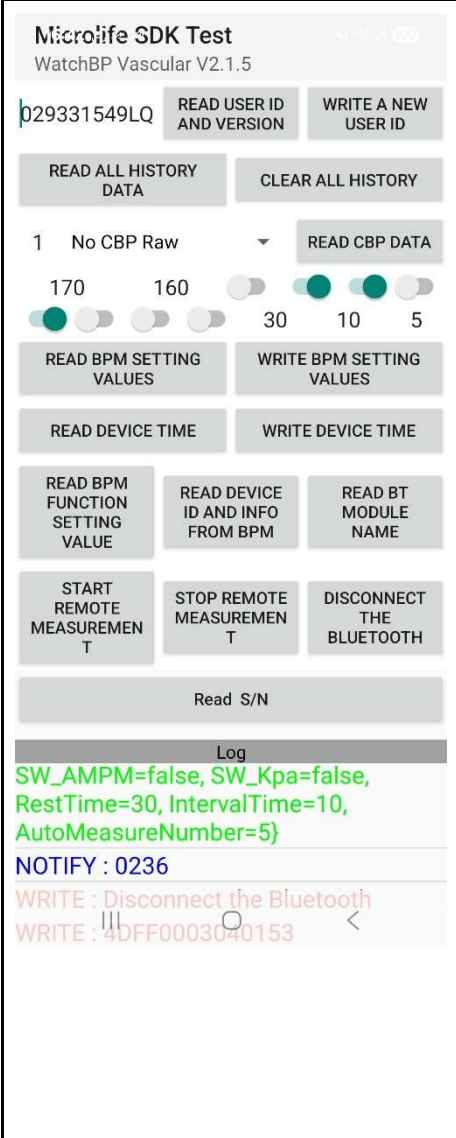
[illegible]

<div> <div> <div>Micr</div> <div>00:59</div> <div>Test</div> </div> <div>WatchBP Vascular V2.1.5</div> </div> <div>Log</div> <div>NOTIFY : 0225</div> <div>WRITE : Read all history data</div> <div>WRITE : 4DFF000300014F</div> <div>NOTIFY : 4D430091000300031C0000000303000000000004</div> <div>NOTIFY : 00000000000000000303000000000034000000000000</div> <div>NOTIFY : 0000010000000000000000AA0196C355FE00030300</div> <div>NOTIFY : 00000000040000000000000000030300000000034</div> <div>NOTIFY : 00000000000000000200000000000000AA0196C355</div> <div>NOTIFY : FE0003030000000000000400000000000000030300</div> <div>NOTIFY : 000000003400000000000000000003000000000000</div> <div>NOTIFY : AA0196C356FE00</div> <div>onResponseReadAllHistorys :</div> <div>DRecord{mode=3,</div> <div>noOfCurrentMeasurement=0,</div> <div>historyMeasuremeNumber=3,</div> <div>currentData=null, MDataWBV=[Curr</div> <div>entAndMDataWBV{R_condition=3,</div> <div>R_systole=3, R_dia=0, R_formulaMAP=0,</div> <div>R_cbp_error=0, R_bp_error=1,</div> <div>R_bodyParts=0, R_oscillometricMAP=0,</div> <div>R_mean_cbp_Data=0, R_pvr_length=0,</div> <div>R_csbp=0, R_cpp=0, L_condition=3,</div> <div>L_systole=3, L_dia=0, L_formulaMAP=0,</div> <div>L_cbp_error=0, L_bp_error=1,</div> <div>L_bodyParts=3, L_oscillometricMAP=0,</div> <div>L_mean_cbp_Data=0, L_pvr_length=0,</div>	<div>CurrentAndMDataWBV{R_condition=3</div> <div>, R_systole=3, R_dia=0,</div> <div>R_formulaMAP=0, R_cbp_error=0,</div> <div>R_bp_error=1, R_bodyParts=0,</div> <div>R_oscillometricMAP=0,</div> <div>R_mean_cbp_Data=0, R_pvr_length=0,</div> <div>R_csbp=0, R_cpp=0, L_condition=3,</div> <div>L_systole=3, L_dia=0,</div> <div>L_formulaMAP=0, L_cbp_error=0,</div> <div>L_bp_error=1, L_bodyParts=3,</div> <div>L_oscillometricMAP=0,</div> <div>L_mean_cbp_Data=0, L_pvr_length=0,</div> <div>L_csbp=0, L_cpp=0, cdia=0,</div> <div>cbpErrorCode=0,</div> <div>currentMeasureNumber=2, hr=0,</div> <div>abi=0.0, ptt=0, pwv=0, height=170,</div> <div>iad_R=0, iad_L=0, afib=0, lowBattery=0,</div> <div>year=2025, month=6, day=24,</div> <div>hour=13, minute=21,</div> <div>abiErrorCode=254, cbpError=0,</div> <div>bpError=0, storeData=0,</div> <div>deviceMode=67, M_Afib=0, M_CBP=0,</div> <div>M_ABI=1, M_PWV=1, M_D_1=1},</div> <div>CurrentAndMDataWBV{R_condition=3</div> <div>, R_systole=3, R_dia=0,</div> <div>R_formulaMAP=0, R_cbp_error=0,</div> <div>R_bp_error=1, R_bodyParts=0,</div> <div>R_oscillometricMAP=0,</div> <div>R_mean_cbp_Data=0, R_pvr_length=0,</div> <div>R_csbp=0, R_cpp=0, L_condition=3,</div> <div>L_systole=3, L_dia=0,</div> <div>L_formulaMAP=0, L_cbp_error=0,</div> <div>L_bp_error=1, L_bodyParts=3,</div> <div>L_oscillometricMAP=0,</div> <div>L_mean_cbp_Data=0, L_pvr_length=0,</div> <div>L_csbp=0, L_cpp=0, cdia=0,</div> <div>cbpErrorCode=0,</div> <div>currentMeasureNumber=3, hr=0,</div> <div>abi=0.0, ptt=0, pwv=0, height=170,</div> <div>iad_R=0, iad_L=0, afib=0, lowBattery=0,</div> <div>year=2025, month=6, day=24,</div> <div>hour=13, minute=22,</div> <div>abiErrorCode=254, cbpError=0,</div> <div>bpError=0, storeData=0,</div> <div>deviceMode=67, M_Afib=0, M_CBP=0,</div> <div>M_ABI=1, M_PWV=1, M_D_1=1}},</div> <div>Setting=28, M_Afib=0, M_CBP=0,</div> <div>M_ABI=1, M_PWV=1, M_d-1=1 }</div>
---	---

## 6.6 Command: Read BPM setting values from BPM

 <p><b>MicroLife SDK Test</b> WatchBP Vascular V2.1.5</p> <p>029331549LQ <b>READ USER ID AND VERSION</b> <b>WRITE A NEW USER ID</b></p> <p><b>READ ALL HISTORY DATA</b> <b>CLEAR ALL HISTORY</b></p> <p>1 No CBP Raw <b>READ CBP DATA</b></p> <p>170 160 30 10 5</p> <p><b>READ BPM SETTING VALUES</b> <b>WRITE BPM SETTING VALUES</b></p> <p><b>READ DEVICE TIME</b> <b>WRITE DEVICE TIME</b></p> <p><b>READ BPM FUNCTION SETTING VALUE</b> <b>READ DEVICE ID AND INFO FROM BPM</b> <b>READ BT MODULE NAME</b></p> <p><b>START REMOTE MEASUREMENT</b> <b>STOP REMOTE MEASUREMENT</b> <b>DISCONNECT THE BLUETOOTH</b></p> <p><b>Read S/N</b></p> <p><b>Log</b></p> <pre>onResponseReadSettingValues : SettingValues{Height=170, HI_infPressure=160, SW_AUTO_hide=false, SW_SEL_silent=true, SW_AUS_Hide=true, SW_AVG_no_include_first=false}</pre>	<ol style="list-style-type: none"> <li>1. This is to perform the function Read BPM setting values from BPM.</li> <li>2. <code>onResponseReadSettingValues :</code>  <code>SettingValue{Height=170,</code>  <code>Hi_infPressure=160,</code>  <code>SW_AUTO_hide=false,</code>  <code>SW_SEL_silent=true,</code>  <code>SW_AUS_Hide=true,</code>  <code>SW_AVG_no_include_first=false,</code>  <code>SW_CBP=true, SW_AFib=false,</code>  <code>SW_AMPM=false, SW_Kpa=false,</code>  <code>RestTime=30, IntervalTime=10,</code>  <code>AutoMeasureNumber=5 }</code></li> </ol>
--	--

## 6.7 Command: Disconnect the Bluetooth

	<ol style="list-style-type: none"> <li>1. Disconnect the Bluetooth</li> <li>2. The command is “4DFF0003040153” and it is sent to device.</li> <li>3. The result is able to observe by the “onConnectionState(ConnectState state)” and its status can be found by the “ConnectState= Disconnect” .</li> <li>4. The “Disconnected” will be displayed on GUI layout. And then, the scanning (discovery) is automatically run to discover devices in the vicinity.</li> </ol>
--	---

