

Demo APP for MicroLifeDeviceSDK (Android)

Table of Contents

Chapter 1	Development Environment
Chapter 2	Entry Point and Bluetooth LE Protocol
Chapter 3	Bluetooth LE Protocol APIs
Chapter 4	BPM APIs
Chapter 5	User Interface of Demo App
Chapter 6	Functionality of Demo App
	Appendix

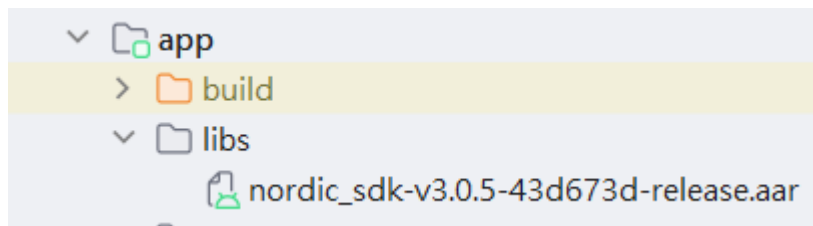
Chapter1 Development Environment

1.1 The supported SDK version is as follow:

```
android {
    namespace "com.mlc.nordic_sdk"
    compileSdkVersion 36

    defaultConfig {
        applicationId "com.mlc.nordic_sdk"
        minSdk 29
        targetSdk 36
        versionCode 1
        versionName "2.2.0"
    }
}
```

1.2 Add the library “nordic_sdk_vxxx-xxxxxxx.aar” into the “libs” directory.



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

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

//nordic scan (sdk required)
implementation libs.scanner

//nordic connect (sdk required)
implementation libs.ble
implementation libs.ble.ktx
implementation libs.ble.common
implementation libs.ble.livedata

//open csv (sdk required)
implementation (libs.opencsv)

//livedata
implementation libs.runtime.livedata
```

Chapter2 Entry Point and Bluetooth LE Protocol

The “ChoseActivity” is the entry point of the sample application. The “BPMAActivity” is dedicated to the device of general Blood Pressure Monitor (Bluetooth LE).

```

        android:theme="@style/Theme.Nordic_SDK" />
    <activity
        android:name=".activity.ChoseActivity"
        android:exported="true"
        android:theme="@style/Theme.Nordic_SDK">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <activity
        android:name=".activity.BPMAActivity"
        android:exported="true"
        android:theme="@style/Theme.Nordic_SDK" />
    <activity
        android:name=".activity.SP02Activity"
        android:exported="true"
        android:theme="@style/Theme.Nordic_SDK" />
    <activity
        android:name=".activity.PFMAActivity"
        android:exported="true"
        android:theme="@style/Theme.Nordic_SDK" />
    <activity
        android:name=".activity.ThermoActivity"
        android:exported="true"
        android:theme="@style/Theme.Nordic_SDK" />
    <activity
        android:name=".activity.BaseActivity"
        android:exported="true"
        android:theme="@style/Theme.Nordic_SDK"/>
    <activity
        android:name=".activity.BGMAActivity"
        android:exported="true"
        android:theme="@style/Theme.Nordic_SDK"/>
    <activity
        android:name=".activity.WBPAActivity"
        android:exported="true"
        android:theme="@style/Theme.Nordic_SDK"/>
</application>

```

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

```
bluetoothManager = BluetoothManager.getInstance( activity = this, listener = this)
```

2.1.1 OnIMBluetoothLEListener: For Bluetooth progress.

```
interface OnIMBluetoothLEListener {
    3 Usages  5+ Implementations
    fun onScanResult(
        device: BluetoothDevice,
        deviceName: String,
        deviceType: DeviceType?,
        macAddress: String?
    )

    5+ Implementations
    fun onConnectionState(connectState: ConnectState)

    5+ Implementations
    fun onConnectionState(connectState: ConnectState, state: Int)

    3 Usages  5+ Implementations
    fun onReceivedBleDataResult(data: List<Byte>, head: Int? = null)

    2 Usages  5+ Implementations
    fun onResponseSWRevision(swRevision: String)

    2 Usages  5+ Implementations
    fun onResponseFWRevision(fwRevision: String)

    2 Usages  5+ Implementations
    fun onResponseHWRevision(hwRevision: String)

    2 Usages  5+ Implementations
    fun onBtStateChanged(isEnable: Boolean)

    2 Usages  5+ Implementations
    fun onRetryFailed(msg: String)
}
```

2.2 Initialize the instance “bpmProtocol”. This is to fulfill retrieving BPM commands and parsing data from the BPM device.

```
// Input protocol key
this.bpmProtocol = BPMProtocol.getInstance(
    sdkid = "",
    bpmType = "${deviceType?.name}",
    listener
)
```

2.2.1 BPMProtocol. OnDataResponseListener: For BPM Protocol progress

```
interface OnDataResponseListener {
    5 Usages 2 Implementations
    fun onResponseBPMReadHistory(dRecord: DRecordBPM?)
    5 Usages 2 Implementations
    fun onResponseBPMReadUserAndVersionData(user: User, versionData: VersionData)
    3 Usages 2 Implementations
    fun onResponseBPMReadDeviceTime(deviceTime: DeviceTime)
    7 Usages 2 Implementations
    fun onResponseBPMReadLastData(
        dRecord: CurrentAndMData?,
        historyMeasurementNumber: Int,
        userNumber: Int,
        MAMState: Int,
        isDataExist: Boolean
    )
    3 Usages 2 Implementations
    fun onResponseBPMReadDeviceInfo(deviceInfo: DeviceInfo)
    5 Usages 2 Implementations
    fun onResponseBPMClearAllHistory(isSuccess: Boolean)
    3 Usages 2 Implementations
    fun onResponseBPMClearLastData(isSuccess: Boolean)
    3 Usages 2 Implementations
    fun onResponseBPMWriteDeviceTime(isSuccess: Boolean)
    5 Usages 2 Implementations
    fun onResponseBPMWriteUserId(isSuccess: Boolean)
    2 Usages 2 Implementations
    fun onResponseBPMCheckTransmitOk(isSuccess: Boolean)
    5 Usages 2 Implementations
    fun onResponseBPMReadSerialNumber(serialNumber: String)
    3 Usages 2 Implementations
    fun onResponseBPMWriteSerialNumber(isSuccess: Boolean)
}
```

Chapter3 Bluetooth LE Protocol APIs

3.1 Instance of Bluetooth LE Protocol

3.1.1 Interface :

	<pre>fun getInstance(activity: Activity, listener: OnIMBluetoothLEListener? = null): BluetoothManager?</pre>
Definition	Initialize Bluetooth LE Protocol
Parameter	activity : name of activity or this listener : get the Bluetooth status
	<pre>bluetoothManager = BluetoothManager.getInstance(activity = this, listener = this)</pre>

3.2 Read and Write command to Bluetooth Device

3.2.1 Interface :

	suspend fun writeCommand(data: ByteArray?)
Definition	write command to Bluetooth Device
Parameter	data : ByteArray command

3.2.2 Delegate :

	<pre>override fun onReceivedBleDataResult(data: List<Byte>, head: Int?)</pre>
Definition	This is to get List<Byte> from Bluetooth device.

3.3 Connection State and Result

3.3.1 Interface :

	<pre>override fun onScanResult(device: BluetoothDevice, deviceName: String, deviceType: DeviceType?, macAddress: String?)</pre>
Definition	This is to get Bluetooth information of devices which discovered in the vicinity.
Parameter	<pre>enum class DeviceType { 4 Usages ALL, MLC_ALL, 6 Usages MLC_BPM3G, MLC_BPM4G, MLC_BPM5G, 4 Usages MLC_Thermo, MLC_SP02, MLC_PF2, 5 Usages MLC_WS, MLC_BGM, MLC_WBP, 1 Usage WAG_ALL, WAG_BPM4G }</pre>

	override fun onConnectionState(connectState: ConnectState)
Definition	The “onConnectionState()” is to monitor the status of connection.
Parameter	<pre>enum class ConnectState { 1 Usage StartScan, StopScan, ScanFinished, Connected, DeviceReady, 4 Usages Disconnect, ConnectFailed, 2 Usages Bonded, BondNone, Bonding, ERROR_133 }</pre>

	override fun onReceivedBleDataResult(data: List<Byte>, head: Int?)
Definition	This is to get List<Byte> from Bluetooth device.

	override fun onResponseSWRevision(swRevision: String)
Definition	This is used to obtain the software revision of the Bluetooth device.

	override fun onResponseFWRevision(fwRevision: String)
Definition	This is used to obtain the hardware revision of the Bluetooth device.

	override fun onBtStateChanged(isEnable: Boolean)
Definition	This is to monitor the state of Enabled or Disabled.

	override fun onRetryFailed(msg: String)
Definition	Return if sending the command fails.

3.4 Device scanning or discovery

3.4.1 Interface :

	fun startScan(scanDeviceType: DeviceType)	
Definition	The “startScan()” is for device scanning or discovery. The result will be shown with the “onScanResult”.	
Parameter	<pre>enum class DeviceType { 4 Usages ALL, MLC_ALL, 6 Usages MLC_BPM3G, MLC_BPM4G, MLC_BPM5G, 4 Usages MLC_Thermo, MLC_SP02, MLC_PF2, 5 Usages MLC_WS, MLC_BGM, MLC_WBP, 1 Usage WAG_ALL, WAG_BPM4G }</pre>	

	fun stopScan()	
Definition	Terminate the scanning process.	

3.5 Disconnection

	fun disconnectGatt()	
Definition	Disconnect device.	

Chapter4 BPM APIs

4.1 Instance of BPM Protocol

4.1.1 Interface :

	<pre>fun getInstance(sdkid: String, bpmType: String, listener: OnDataResponseListener? = null): BPMProtocol?</pre>
Definition	Initialize BPM Protocol
Parameter	<p>sdkid : SDK ID of designated device</p> <p>bpmType : Pass the deviceType parameter from scanResult.</p> <p>listener : This is to obtain the parsed BPM information.</p>
	<pre>// Input protocol key this.bpmProtocol = BPMProtocol.getInstance(sdkid = "", bpmType = "\${deviceType?.name}", listener)</pre>

4.2 Read all history data from BPM :

4.2.1 Interface :

	fun readAllHistory(selectUser: Long): ByteArray?
Definition	Read all history data from the BPM.

4.2.2 Delegate :

	fun onResponseBPMReadHistory(dRecord: DRecordBPM?)
Parameter	<pre> data class DRecordBPM(var mode: Int?, var noOfCurrentMeasurement: Int? = null, var historyMeasuremeNumber: Int? = null, var userNumber: Int? = null, var mamState: Int? = null, var average: Int? = null, var currentData: List<CurrentAndMData?>? = null, var mData: List<CurrentAndMData?>? = null, var measureMode: Boolean? = null) { </pre>

4.3 Clear all history data from BPM :

4.3.1 Interface :

	fun clearAllHistory(selectUser: Long): ByteArray?
Definition	Clear all history data from BPM

4.3.2 Delegate :

	fun onResponseBPMClearAllHistory(isSuccess: Boolean)
Parameter	isSuccess : True of False

4.4 Disconnect the Bluetooth with BPM :

4.4.1 Interface :

	fun disconnect(): ByteArray
Definition	Disconnect the Bluetooth with BPM

4.4.2 Delegate :

	override fun onConnectionState(connectState: ConnectionState)
Definition	The "onConnectionState()" is to monitor the status of connection.
Parameter	<pre>enum class ConnectionState { 1 Usage StartScan, StopScan, ScanFinished, Connected, DeviceReady, 4 Usages Disconnect, ConnectFailed, 2 Usages Bonded, BondNone, Bonding, ERROR_133 }</pre>

4.5 Read last one data from BPM :

4.5.1 Interface :

	<code>fun readLastData(): ByteArray</code>
Definition	Read the last measurement data.

4.5.2 Delegate :

	<pre> fun onResponseBPMReadLastData(dRecord: CurrentAndMData?, historyMeasurementNumber: Int, userNumber: Int, MAMState: Int, isDataExist: Boolean) </pre>
Parameter	<p>CurrentAndMData dRecord: last 1 data</p> <p>historyMeasuremeNumber : History Measurement Number.</p> <p>The rande of Memory Set = 0 ~ 255</p> <p>userNumber: User Number: User1 = 1, User2 = 2, Guest = 3</p> <p>MAMState: MAM state: 0=MAM disable, 1=Weight off, 2=Weight on, 3=Light off, 4=Light on.</p> <p>isNoData: True or False</p>

4.6 Write a new user ID & Age to BPM :

4.6.1 Interface :

	<pre>fun writeUserId(selectUser: Long, id: String, age: Long): ByteArray?</pre>
Definition	Write a new user ID & Age to BPM

4.6.2 Delegate :

	<pre>fun onResponseBPMWriteUserId(isSuccess: Boolean)</pre>
Parameter	isSuccess : True of False

4.7 Clear last one data of BPM :

4.7.1 Interface :

	<pre>fun clearLastData(): ByteArray</pre>
Definition	Clear last one data of BPM

4.7.2 Delegate :

	<pre>fun onResponseBPMClearLastData(isSuccess: Boolean)</pre>
Parameter	isSuccess : True of False

4.8 Read user ID and version data from BPM :

4.8.1 Interface :

	<code>fun readUserAndVersionData(): ByteArray</code>
Definition	Read user ID and version data from BPM.

4.8.2 Delegate :

	<pre>fun onResponseBPMReadUserAndVersionData(user: User, versionData: VersionData)</pre>
Parameter	User : User Info VersionData : Device Info

4.9 Solve Data Result

4.9.1 Interface :

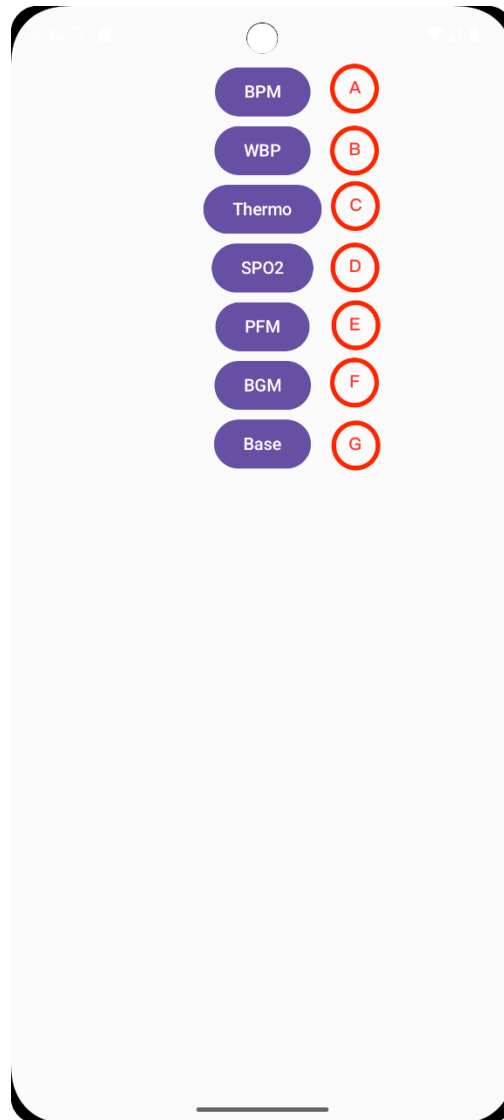
	<code>fun solveDataResult(data: String)</code>
Definition	Parses the List<Byte> received from the Bluetooth device.

	<code>fun solveDataResult(data: List<Byte>)</code>
Definition	Parses the List<Byte> received from the Bluetooth device.

Chapter5 User Interface of Demo App

5.1 Getting Started :

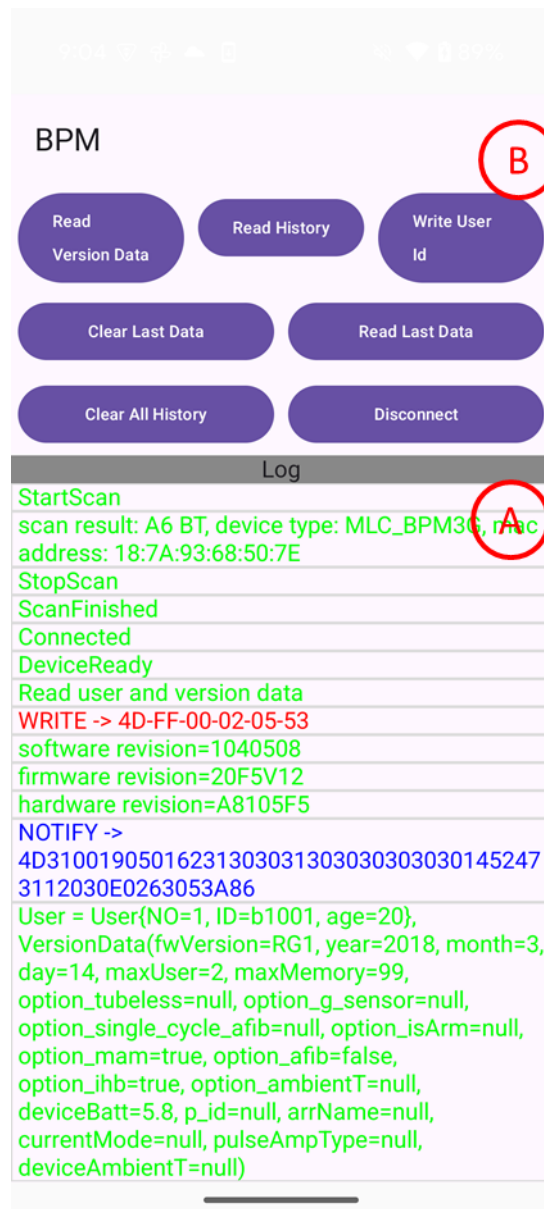
Start the app and then select the button “Blood Pressure ” / “A” to communicate with the designate device BPM.



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. If not, the “bindingDevice” can be used to run bonding process.

5.3 GUI Layout :



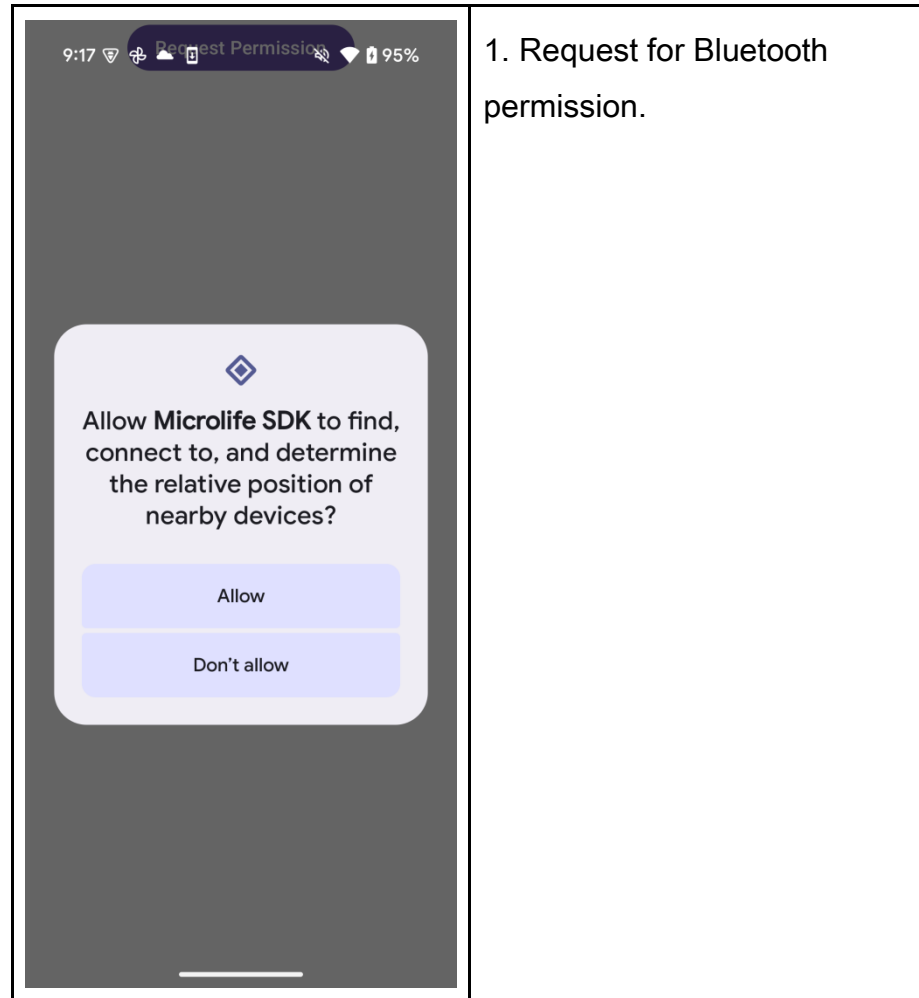
5.3.1 Region A : The log window is used to display information about communication handshake between App and device.

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

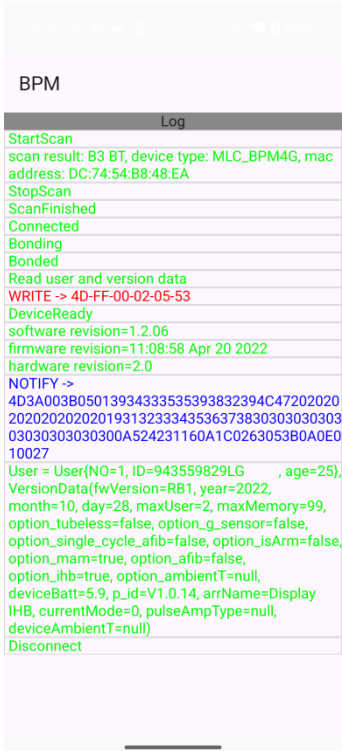
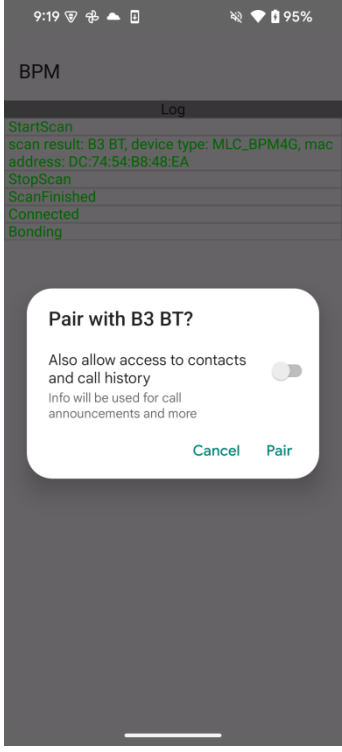
5.3.3 Refer to “BPMActivity” from the demo application (sample code) to get more detailed.

Chapter6 Functionality of Demo App

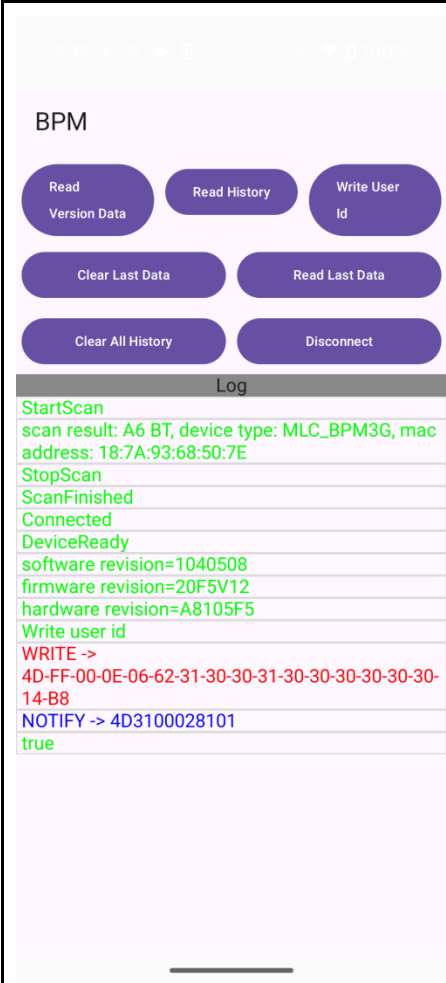
6.1 Bluetooth authorization :



6.2 Pairing / Bonding :

	<p>1. There is a message to confirm the pairing bonding procedure between device and cellphone if they haven't bonded yet.</p> <p>2. Once the procedure is done, choose any function/ command to do communication with BPM device.</p> <p>3. The green part is from "onScanResult".</p>
	

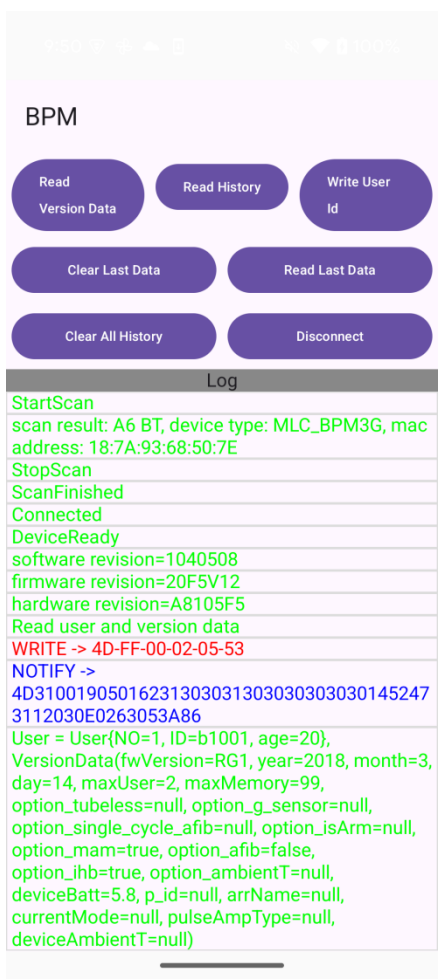
6.3 Command: Write a new user ID to BPM



The screenshot shows the BPM app interface. At the top, there's a status bar with signal and battery icons. Below it, the title 'BPM' is displayed. The main area contains several buttons: 'Read Version Data', 'Read History', 'Write User Id', 'Clear Last Data', 'Read Last Data', 'Clear All History', and 'Disconnect'. Below these buttons is a 'Log' section with a list of events. The events are color-coded: green for status messages, red for commands, and blue for notifications. The log shows a sequence of events including 'StartScan', 'scan result: A6 BT, device type: MLC_BPM3G, mac address: 18:7A:93:68:50:7E', 'StopScan', 'ScanFinished', 'Connected', 'DeviceReady', 'software revision=1040508', 'firmware revision=20F5V12', 'hardware revision=A8105F5', 'Write user id', 'WRITE -> 4D-FF-00-0E-06-62-31-30-30-31-30-30-30-30-30-14-B8', 'NOTIFY -> 4D3100028101', and 'true'.

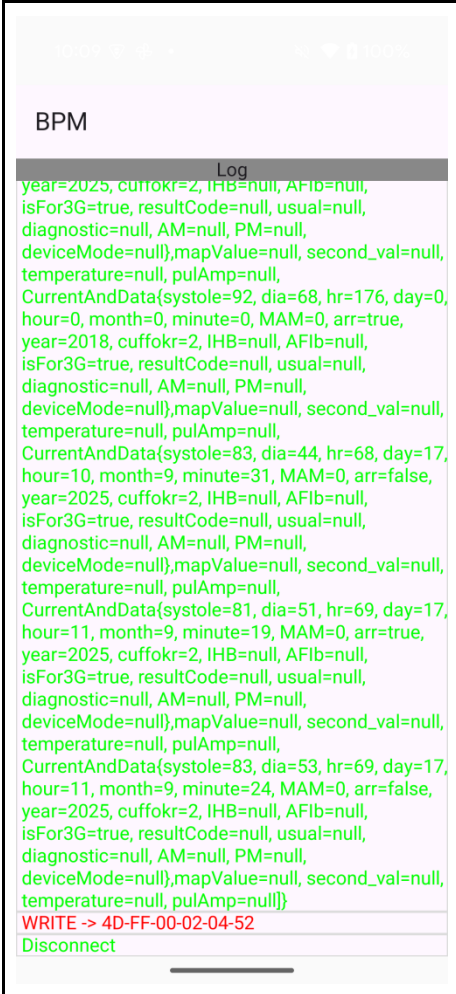
1. The command “WRITE USER” is to write a new user ID to BPM.
2. The log “WRITE : write user” and “Write : userID:b1001” are indicated that the App sends a command with an user ID. The ID is made up of ASCII code.
3. The log “BPM:WriteUser -> isSuccess = true” means that the writing/ sending procedure is successful.
4. The red part is the command and communication protocol that is sent to device. The blue part is notification with the raw data from BPM via Bluetooth.

6.4 Command: Read user ID and version data from BPM

 <p>The screenshot shows the BPM application interface. At the top, there's a status bar with 'BPM' and a signal strength indicator. Below that, there are several buttons: 'Read Version Data', 'Read History', 'Write User Id', 'Clear Last Data', 'Read Last Data', 'Clear All History', and 'Disconnect'. A 'Log' section is visible at the bottom, displaying a series of green text logs. The logs include 'StartScan', 'scan result: A6 BT, device type: MLC_BPM3G, mac address: 18:7A:93:68:50:7E', 'StopScan', 'ScanFinished', 'Connected', 'DeviceReady', 'software revision=1040508', 'firmware revision=20F5V12', 'hardware revision=A8105F5', 'Read user and version data', 'WRITE -> 4D-FF-00-02-05-53', 'NOTIFY ->', a long hexadecimal string '4D3100190501623130303130303030301452473112030E0263053A86', and a detailed JSON-like structure for 'User' and 'VersionData'.</p>	<p>1. The command “Read user ID and version data” is to get user ID and device information as below.</p> <p>2. The log “BPM : ReadUserAndVersionData -> user = User{NO=1, ID='b1001', age=20} VersionData : VersionData{year=2018, month=3, day=14, maxUser=2, maxMemory=99, optionIHB=true, optionAfib=false, optionMAM=true, optionAmbientT=false, optionTubeless=false, optionDeviceID=false, deviceBatteryVoltage=5.8, FWName='RG1'}” is included ID, Age and Device information (Battery voltage, Firmware version and so on).</p>
--	--

	<p>pulse. For instance, the above-mentioned reading is DateTime 2021/1/19, Sys 118, Dia 77, Pulse 75.</p>
--	---

6.6 Command: Disconnect the Bluetooth

 <p>The screenshot shows the BPM application interface. At the top, there's a status bar with a signal strength indicator and the time 10:52. Below that, the text 'BPM' is displayed. A 'Log' section contains a long list of data points, including year, cuffokr, IHB, AFib, isFor3G, resultCode, usual, diagnostic, AM, PM, deviceMode, mapValue, second_val, temperature, pulAmp, CurrentAndData, hour, month, minute, MAM, arr, and year. The log entries are color-coded: green for most data, red for 'WRITE -> 4D-FF-00-02-04-52', and green for 'Disconnect'.</p>	<ol style="list-style-type: none"> 1. The button “DISCONNECT” is to terminate the connection of Bluetooth. 2. The “DISCONNECT” sends a disconnected command to BPM, and then the disconnection will be executed by BPM.
--	---

Appendix

CurrentAndMData Structure / Parameter				
Property	Device	Type	Value	Description
AA=0	WBP	Integer	0, 1	Anti-artifact Detected
ABPM=0	WBP	Integer	0, 1	Ambulatory Blood Pressure Monitoring
AFIb=false	3G/4G	Boolean	false , true	Atrial Fibrillation Detection
AM=false	WBP	Boolean	false , true	Data measured at night with diagnostic mode (e.g., 4:00~12:00)
CPP=0	WBP	Integer	0 ~ 255	Central Pulse Pressure
CSBP=0	WBP	Integer	0 ~ 255	Carotid Systolic Blood Pressure
IHB=false	3G/4G	Boolean	false , true	Irregular Heartbeat Detection
LB=0	WBP	Integer	0, 1	Low Battery
MAM=0	3G/4G	Integer	0 ~ 3	Microlife Aerge Mode
MAP=0	WBP	Integer	0 ~ 255	Mean Arterial Pressure
MCBP=0	WBP	Integer	0 ~ 65535	Mean Central Blood Pressure
PM=false	WBP	Boolean	false , true	Data measured in the morning with diagnostic mode

				(e.g., 18:00~24:00)
PVR=0	WBP	Integer	0 ~ 65535	Pulse Variation Rate
SM=0,	WBP	Integer	0, 1	Start of a Manual Measurement
arr=false	3G/4G	Boolean	false , true	Detection of PAD or Afib
condition=0	WBP	Integer	0 ~ 24	Condition
Cuffokr=0	3G/4G	Integer	0, 1	Whether the wristband is tight: 3G detection mode: 2 - No detection, 4G detection mode: 0 - No tightness, 1 - There is tightness
day=7	WBP, 3G/4G	Integer	0 ~ 31	Date
deviceMode=58	WBP, 3G/4G	Integer	0x31 3G BPM, 0x3A 4G BPM, 0X51 WBP HomeA	Device type
dia=69	WBP, 3G/4G	Integer	0 ~ 255	Diastolic
diagnostic=false	WBP	Boolean	false , true	Data measured in diagnostic mode
errorCode=0	WBP	Integer	0 ~ 255	Error Code
hour=10	WBP, 3G/4G	Integer	0 ~ 24	Hour

hr=82	WBP, 3G/4G	Integer	0 ~ 255	Pulse
indexYear=0	WBP	Integer	0 ~ 99	offset Year
isFor3G=false	3G/4G	Boolean	false , true	is for 3G
minute=12	WBP, 3G/4G	Integer	0 ~ 59	Minute
month=2,	WBP, 3G/4G	Integer	1 ~ 12	Month
resultCode=0	WBP	Integer	0x01, 0x02, 0x03, 0x05, 0x42, 0x50, 0x51, 0x52	Result Code
systole=114	WBP, 3G/4G	Integer	0 ~ 255	Systolic
usual=false	WBP	Boolean	false , true	Data measured in usual mode
year=2024	WBP, 3G/4G	Integer	WBP: 2000~ 2050, 3G/ 4G: 2000 ~ 2048	Year