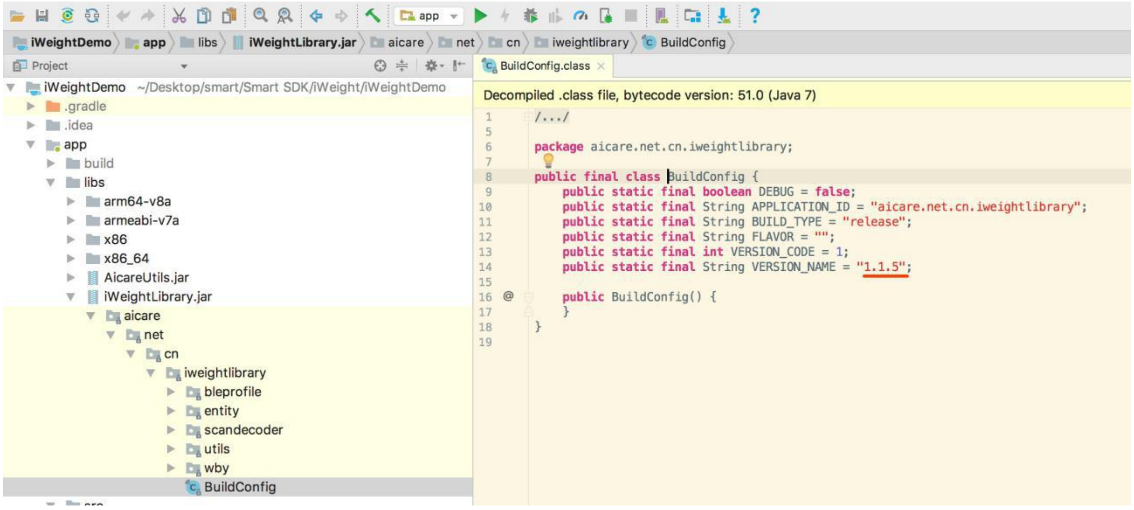
Body fat scale SDK instructions

Version	Update Time	Author	Update information
1.0	2017/2/15	Suzy	Init version
1.0.1	2017/7/31	Suzy	Fix some data conversion bugs
1.0.2	2017/9/11	Suzy	The type in the instruction is set according to the type of device actually connected.
1.0.3	2017/11/16	Suzy	Code optimization
1.0.4	2018/7/12	Suzy	Compatible with broadcast scales
1.0.5	2018/1/15	Suzy	Compatible with new protocols
1.0.6	2018/2/25	Suzy	Function optimization
1.0.7	2018/2/28	Suzy	Package decimal point class
1.0.8	2018/3/14	Suzy	Fix bugs in body age errors
1.0.9	2018/4/9	Suzy	Compatible with BM09 protocol
1.1.1	2018/5/10	Suzy	Compatible with kg/lb indexing protocol
1.1.2	2018/5/11	Suzy	Solve the bug that the st unit data is inconsistent with the scale display
1.1.3	2018/6/22	Suzy	Solve bugs that are sometimes not available for decimal point
1.1.4	2018/7/7	Suzy	Compatible with BM15 protocol
1.1.5	2018/7/20	Suzy	Compatible algorithm ID protocol
1.1.7	2019/04/30	Stan	Added BM15 body fat data calculation algorithm
1.1.8	2019/06/6	Stan	修改原始数据转lb ,st计算错误问题
1.1.9	2019/12/19	Xing	Modify the original data to lb, st calculation error
1.2.0	2019/1/17	Xing	Updated and optimized Bluetooth library
1.2.1	2020/3/19	Xing	Increase body fat data calculation method and fat-free weight algorithm, etc.
1.2.2	2020/03/26	Xing	Update jar to fix known bugs
1.2.3	2020/04/02	Xing	Increase key, secret check

目录

A. Precautions :

1. The latest version of iWeightLibrary.jar is 1.1.5. (2018/8/16)



Decompiled .class file, bytecode version: 51.0 (Java 7)

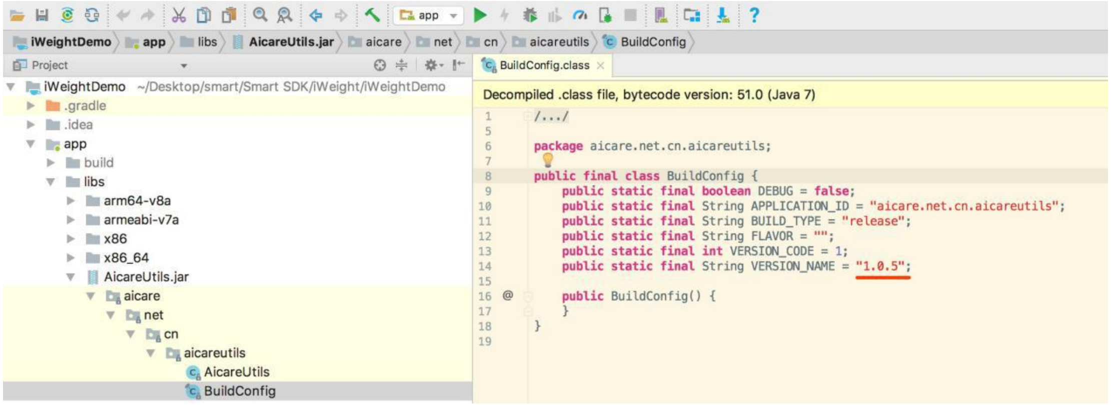
```
/.../

package aicare.net.cn.iweightlibrary;

public final class BuildConfig {
    public static final boolean DEBUG = false;
    public static final String APPLICATION_ID = "aicare.net.cn.iweightlibrary";
    public static final String BUILD_TYPE = "release";
    public static final String FLAVOR = "";
    public static final int VERSION_CODE = 1;
    public static final String VERSION_NAME = "1.1.5";

    public BuildConfig() {
    }
}
```

2. The latest version of AicareUtils.jar is 1.0.5. (2018/8/16)



Decompiled .class file, bytecode version: 51.0 (Java 7)

```
/.../

package aicare.net.cn.aicareutils;

public final class BuildConfig {
    public static final boolean DEBUG = false;
    public static final String APPLICATION_ID = "aicare.net.cn.aicareutils";
    public static final String BUILD_TYPE = "release";
    public static final String FLAVOR = "";
    public static final int VERSION_CODE = 1;
    public static final String VERSION_NAME = "1.0.5";

    public BuildConfig() {
    }
}
```

3. If targetSdkVersion>22 needs to dynamically obtain location permissions(some phones need to turn on GPS).

4. Users/units can be synchronized only after receiving STATE_INDICATION_SUCCESS.

5. Please synchronize the user after receiving STATE_INDICATION_SUCCESS, otherwise the body fat data will not be obtained.

6. The unit only supports 4 kinds (kg, lb, st, jin) at most. Please refer to the scale for specific units.

7. The weight data returned by the scale end is not converted. To convert, use the AicareBleConfig.getWeight(double weight, byte unit, DecimalInfo decimalInfo) method.

8. The weight data returned by the scale is not converted. To convert, use the AicareBleConfig.getWeight (double weight, byte unit, DecimalInfo decimalInfo) method.

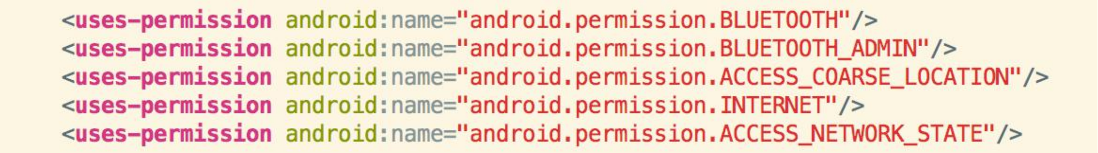
9. The recommended interval for sending data is 200 milliseconds or more.

B. Conditions of Use :

1. The lowest version of android4.3(API 18).
2. The Bluetooth version used by the device requires 4.0 and above.

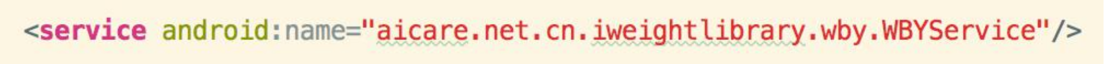
C. Start using :

1. Set minSdkVersion: minSdkVersion needs to be greater than or equal to 18.
2. Add related permissions:



```
<uses-permission android:name="android.permission.BLUETOOTH"/>
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
```

3. Registration service:



```
<service android:name="aicare.net.cn.iweightlibrary.wby.WBYService"/>
```

Initial registration in the application, key and secret, please go to our SDK website to register

AiFitSDK.getInstance (). Init (Context, "key", "secret");

4. The Activity that needs to get the device data inherits the BleProfileServiceReadyActivity and overrides the related methods.

D. Discovery device:

1. Determine if the device supports BLE:

Call ensureBLESupporte() in BleProfileServiceReadyActivity. If true, the device supports BLE.

2. Bluetooth status:

The Bluetooth switch will trigger bluetoothStateChanged(int state) in BleProfileServiceReadyActivity.

i. Bluetooth is turned on: BluetoothAdapter.STATE_ON

ii. Bluetooth is off: BluetoothAdapter.STATE_OFF

iii. Bluetooth is turning on: BluetoothAdapter.STATE_TURNING_ON

iv. Bluetooth is turning off: BluetoothAdapter.STATE_TURNING_OFF

3. Start scanning:

   If the device supports BLE, call startScan() in BleProfileServiceReadyActivity to start scanning.

4. Stop scanning:

   You can stop the scan by calling stopScan() in BleProfileServiceReadyActivity when you don't need to scan.

F. Device Type:

The device is mainly divided into the following two categories. The device type can be obtained according to BroadData.getDeviceType().

1. Broadcast scale (weight data is obtained by parsing Bluetooth broadcast data, and Bluetooth broadcast data can be obtained according to BroadData.getSpecificData()):

   1.1 Weight broadcast scale (deviceType=0), weight broadcast scale with temperature (deviceType=1):

   Call AicarConfig.getWeightData(byte[] b) to get the WeightData.

   1.2 BM09 broadcast scale (deviceType=9)

   Call AicareConfig.getBm09Data(String address, byte[] specialData) to get the BM09Data。

   1.3 BM15 broadcast scale (deviceType=15)

   Call AicareConfig.getBm15Data(String address, byte[] specialData) to get the BM15Data.

2. Connection scale:

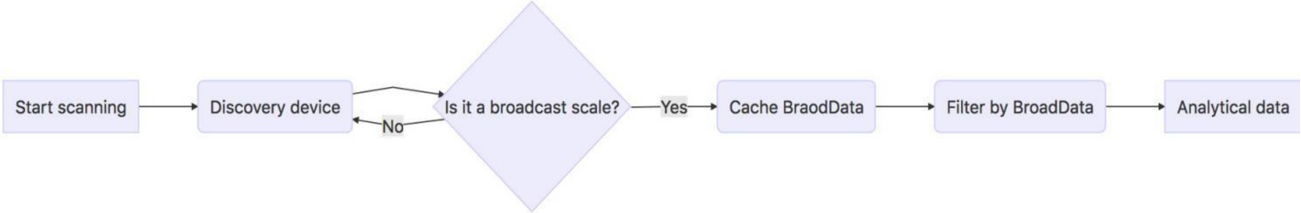
   2.1 Calculate body fat data on the scale

   The body fat data is returned by BleProfileServiceReadyActivity.onGetFatDat(boolean isHistory, BodyFatData bodyFataData) on the premise of successful synchronization of the user and impedance measurement.

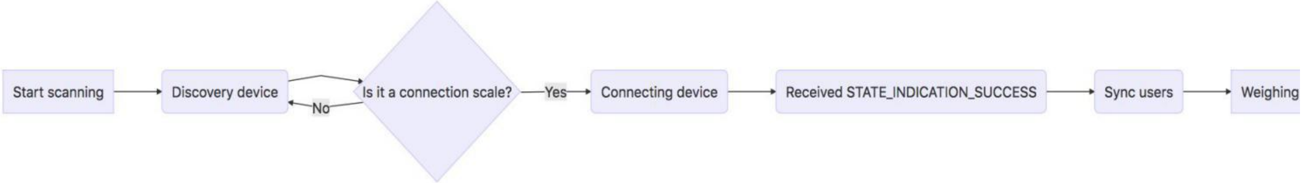
   2.2 Scale-end return algorithm ID (calculates body fat data from app)
   On the premise that the impedance measurement is successful, the algorithm sequence information is returned by BleProfileServiceReadyActivity.onGetAlgorithmInfo(AlgorithmInfo algorithmInfo).

G. Flow Description:

1. Broadcast scale:



2. Connection scale:



H. Communicate with the device:

1. Connect the device:
   a) When scanned to the device, getAicareDevice(BroadData broadData) in BleProfileServiceReadyActivity is triggered. (If you have multiple devices, you need to handle it yourself)
   b) After scanning to the device that needs to be connected, call startConnect (String address) in BleProfileServiceReadyActivity to connect to the device.
2. Connection Status (#onStateChanged(String deviceAddress, int state)
   a) Connection succeeded: BleProfileService.STATE_CONNECTED。
   b) Disconnect: BleProfileService.STATE_DISCONNECTED。
   c) Discovery service: BleProfileService.STATE_SERVICES_DISCOVERED。
   d) Indication success: BleProfileService.STATE_INDICATION_SUCCESS。
   e) Connection timed out: BleProfileService.STATE_TIME_OUT。
   f) Connecting: BleProfileService.STATE_CONNECTING。
3. Connection error (#onError(String msg, int errorCode)
4. The app sends data to the device:
   a) After successfully connecting to the device, you can get the bridge binder that communicates with the device through the onServiceBinded(E binder) method in BleProfileServiceReadyActivity. (See Demo for details)

b) After getting WBYService.WBYBinder, you can call the corresponding method in WBYBinder to communicate with the device.

WBYBinder includes the following methods:

1) Sync user: syncUser(User user);

2) Sync unit: syncUnit(byte unit);

5. The device sends data to the app:

a) After get STATE_INDICATION_SUCCESS, the data sent by the device to the app will trigger the following corresponding methods:

i. Change and stable weight data and temperature (only AC03 supports temperature):

onGetWeightData(WeightData weightData);

ii. Status information returned by the device:

onGetSettingStatus(@AicareBleConfig.SettingStatus int status);

iii. Information such as the Bluetooth version, date, time, and impedance returned by the device:

onGetResult(int index, String result);

iv. The body fat data returned by the device (only need to get the data whose isHistory is false):

onGetFatData(boolean isHistory, BodyFatData bodyFatData);

v. The number of decimal point returned by the device:

onGetDecimalInfo(DecimalInfo decimalInfo);

vi. Algorithm ID information returned by the device:

onGetAlgorithmInfo(AlgorithmInfo algorithmInfo);

vii. The device returns impedance. If there is no body fat data, you can call the algorithm in the SDK to obtain body fat data. After getting BodyFatData in iv, call the algorithm to calculate the cn.net.aicare.algorithmutil.BodyFatData object data.

AicareBleConfig.getBodyFatData(AlgorithmUtil.AlgorithmType.TYPE_AICARE, bodyFatData.getSex(), bodyFatData.getAge(), Double.valueOf(ParseData.getKgWeight(bodyFatData.getWeight(), bodyFatData.getDecimalInfo())),

```
bodyFatData .getHeight(), bodyFatData.getAdc());
```

viii. If you need to obtain 6 additional physical indicators such as fat-free weight and weight control amount, please call the algorithm provided by SDK to calculate MoreFatData object.

AicareBleConfig.getMoreFatData(int sex, int height, double weight, double bfr, double rom, double pp)

6. Disconnect:

You can disconnect by calling disconnect() in WBYService.WBYBinder.

I. Object description (aicare.net.cn.iweightlibrary.entity) :

1. AlgorithmInfo(Algorithm ID information)

Type	Param Name	Description
double	weight	Weight
int	algorithmId	Algorithm ID
int	adc	Impedance value
DecimalInfo	decimalInfo	Decimal point

2. BM09Data

Type	Param Name	Description
int	agreementType	Agreement type
int	unitType	Unit type
DecimalInfo	decimalInfp	Decimal point
double	weight	Weight
int	adc	Impedance value
double	temp	Temperature
int	algorithmId	Algorithm ID
~~int~~	~~did~~	~~(Currently useless)~~
String	bleVersion	Bluetooth version
int	bleType	Device type(0x09)
String	address	Device address
long	timeMillis	Measurement timestamp
boolean	isStable	Is the data stable

3. BM15Data

Type	Param Name	Description
String	version	Bluetooth version
int	agreementType	Agreement type
int	unitType	Unit type
double	weight	Weight
int	adc	Impedance value
double	temp	Temperature (if temp=6553.5, the scale does not support temperature)

int	algorithmId	Algorithm ID
~~int~~	~~did~~	~~(Currently useless)~~
int	bleType	Device type(0x15)
String	address	Device address

4. BodyFatData

Type	**Param Name**	**Description**
String	date	Measurement date
String	time	Measurement time
double	weight	Weight
double	bmi	Body mass index
double	bfr	Body fat rate
double	sfr	Subcutaneous fat rate
int	uvi	Visceral fat
double	rom	Rate of muscle
double	bmr	Basal metabolic rate
double	bm	Bone Mass
double	vwc	Moisture rate
double	bodyAge	Physical bodyAge
double	pp	Protein percentage
int	number	ID
int	sex	Sex(1, male; 2, female)
int	age	Age
int	height	Height
int	adc	Impedance value

5. BroadData

Type	**Param Name**	**Description**
String	name	Device name
String	address	Device address
boolean	isBright	Is the scale bright
int	rssi	Signal value
byte[]	specificData	Broadcast data
int	deviceType	Device type

6. DecimalInfo

Type	Param Name	Description
int	sourceDecimal	Source data decimal point
int	kgDecimal	kg decimal point
int	lbDecimal	lb decimal point
int	stDecimal	st decimal point
int	kgGraduation	kg graduation
int	lbGraduation	lb graduation

7. User

Type	Param Name	Description
int	id	ID
int	sex	Sex(1, male; 2, female)
int	age	Age
int	height	Height
int	weight	Weight (not required in syncUser)
~~int~~	~~adc~~	~~Impedance value(deprecated)~~

8. WeightData

Type	Param Name	Description
int	cmdType	Command type (1,change; 2, stable; 3,impedance measurement)
double	weight	Weight
double	temp	Temperature (if the temperature is Double.MAX_VALUE, the scale does not support temperature)
DecimalInfo	decimalInfo	Decimal point information
int	adc	Impedance value
int	algorithmType	Algorithm ID
int	unitType	Unit type
int	deviceType	Device type