Security：	Confidential	*Version：*	V1.0
Scope：	Customer	*Status：*	Released

HC03 Flutter SDK API Guide

File No.	LT-SW-24062401	*Type*	RD
Written：	*Chenzq*	*Date：*	*2024-6-24*
Check：	*Yangxd*	*Date：*	*2024-6-24*
Approval：	*Tangzp*	*Date：*	*2024-6-24*

LINKTOP

Linktop Technology Co., Ltd

■ Records

Version	*Date*	*Remarks*
V1.0	2024.06.24	First version

■ Table of Contents

1. Introduction

1.1. Engineering Development Description

1. Bluetooth outsourcing, unpacking, and data processing are encapsulated in the lib/src/core directory, and the external API is in the hc03_sdk_impl.dart file
2. The Bluetooth library is encapsulated in the lib/src/bluetooth directory, and the external API is in the bluetool_manager.dart file
3. The call to the function is located in the lib/src/pages_data/health_data.dart file.

1.2. Integration Steps

1.2.1. Android

1. Porting the jar package of Android/app/libs/NskAlgoSdk.jar and the Java files in the Android/app/src/main/Java/com/ecg folder, as well as the So library and Android/app/src/main/kotlin/com/example/smart_flutter/EcgPlugin.kt files in the Android/app/src/main/jniLibs folder to your project
2. Initialize EcgPlugin in the configureFlutterEngine method of MainActivity in your project Android. Please refer to Android/app/src/main/kotlin/com/example/smart_flutter/MainActivity.kt

   Remember to add permission references to Android/app/src/main/AndroidManifest.xml in AndroidManifest.xml

   Please refer to the android/app/build.gradle file for the dependency on NskAlgoSdk. jar and the setting of the jniLibs path that needs to be added in the android/app/build.gradle file.

1.2.2. iOS

1. Porting the files and .a library from the ios/Runner/libNSKAlgoSDECG folder, as well as the ios/Runner/SDK Health Monitor.h and ios/Runner/SDK Health Monitor. m, to your project.

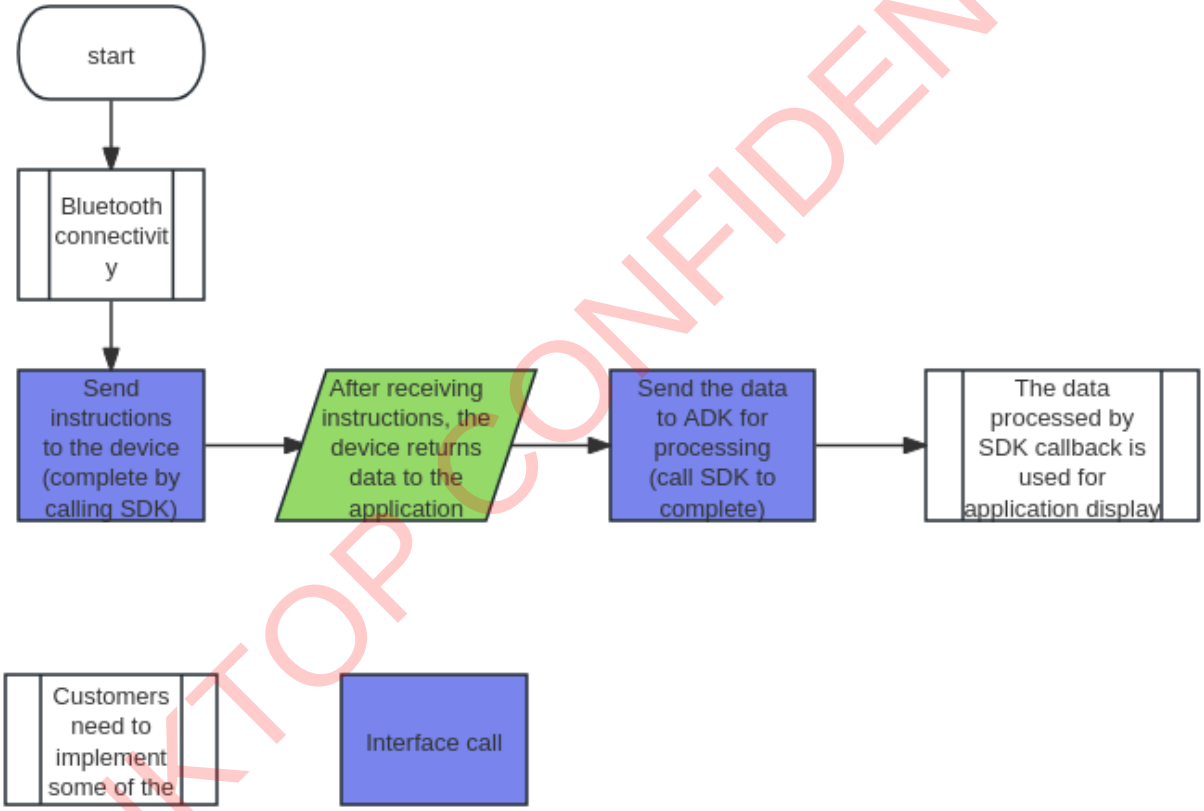
   Note that the .a static library needs to be added in Target ->Build Phases ->Link Binary With Libraries in xcode, and then in Target ->Build Setting ->Other Link Flags, add - all'load (the .a library is located in libNSKAlgoSDECG).
2. To initialize SDKHealthMonitor in ios/Runner/AppDelegate.h and ios/Runner/AppDelegate.m, please refer to the demo for ios/Runner/AppDelegate. h and ios/Runner/AppDelegate.m.

2. API Description

2.1. Function Description

SDK establishes a connection with the device through BLE. You can easily communicate with the device through the SDK and extract services such as electrocardiogram, blood oxygen, and body temperature data provided by the device.

2.2. Flow Chart



2.3. External Interface

2.3.1. Initialization

Initialize where you need to use interfaces

```
Hc03Sdk sdk=Hc03Sdk.getInstance();
```

2.3.2. Start

1) Function

startDetect(Detection detection);

Call startDetect to start the test interface after Bluetooth connection

2) Parameter

- Detection.BT: Temperature
- Detection.OX: Blood oxygen
- Detection.ECG: Electrocardiogram
- Detection.BP: Blood pressure
- Detection.BATTERY: Battery
- Detection.BG: Blood sugar

3) Sample Code

```
void startDetect(Detection detection) async {
  var data = sdk.startDetect(detection);
  await blueToothManager.write(WRITE_UUID, data.toList());
}
```

2.3.3. Stop

1) Function

stopDetect(Detection detection);

Call stopDetect to stop the testing interface

2) Parameter

- Detection.BT: temperature
- Detection.OX: Blood oxygen
- Detection.ECG: Electrocardiogram
- Detection.BP: Blood pressure
- Detection.BG: Blood sugar

3) Sample Code

```
void stopDetect(Detection detection) async {
  var data = sdk.stopDetect(detection);
  await blueToothManager.write(WRITE_UUID, data.toList());
}
```


2.3.4. Callback

1) Function

parseData(data);

2) Parameter

data: Bluetooth callback data

3) Sample Code

```
blueToothManager.notifityWriteListener(NOTIFY_UUID, (data) async {
  sdk.parseData(data);
}, onError: (error) {
  debugPrint("notifityWriteListener error=$error");    "notifity": 
});
```

2.3.5. ECG

1) Function

getEcgData;

2) Callback data type

- Wave: Data used for drawing waveforms
- HR: Heart rate data
- Mood Index: Mood Index

  1-20: chill

  21-40: relax

  41-60: balance

  61-80: excitation

  81-100: Excitement/Anxiety/Excitement

- RR: Peak to peak value
- HRV: Heart rate variability
- RESPIRATORY RATE：Respiratory rate
- touch: Finger detection

3) Sample Code


```
final ecgStream = sdk.getEcgData();
ecgStream.listen((data) {
  if (data is EcgData) {
    var message = data.ecg;
    switch (message["type"]) {
      case "wave":
        waveData.add(message["data"] as int);
        if (waveData.length > 2200) {
          waveData.removeAt(0);
        }
        ecgUpdate.value = !ecgUpdate.value;
        break;
      case "HR":
        hr.value = message["value"];
        break;
      case "Mood Index":
        if (hrv.value != 0) {
          stopEcg();
        }
        mood.value = message["value"];
        break;
      case "RR":
        if (message["value"] > maxRR.value) {
          maxRR.value = message["value"];
        } else if (message["value"] < minRR.value
```

2.3.6. Blood oxygen

1) Function

getBloodOxygen;

2) Callback data type

- BloodOxygenData:

bloodOxygen: Blood oxygen

heartRate: heart rate

- FingerDetection: Finger detection
- BloodOxygenWaveData: Draw waveform data

3) Sample Code


```
final stream = await sdk.getBloodOxygen();
streamSubscription = stream.data.listen((Hc03BaseMeasurementData data) {
  if (data is BloodOxygenData) {
    bloodOxygen.value = data.bloodOxygen;
    heartRate.value = data.heartRate;
  } else if (data is FingerDetection) {
    debugPrint("fingerDetection=${data.isTouch}");
  } else if (data is BloodOxygenWaveData) {
    hrWaveList.add(data.red.wave.toInt());
    if (hrWaveList.length > 400) {
      hrWaveList.removeAt(0);
    }
    hrWaveUpdate.value = !hrWaveUpdate.value;
  }
}, onError: (Object error, StackTrace stackTrace) {
  if (error is AppException) {
    debugPrint("getTemperature exception=${error.message}");
  }
});
stream.stop.listen((Hc03BaseMeasurementData data) {
  if (data is StopMeasure) {
    stopBloodOxygen();
  }
});
```

2.3.7. Glucose

1) Function

getBloodGlucoseData;

2) Callback data type

- BloodGlucoseSendData:Data to be sent to the device
- BloodGlucosePaperState: Blood glucose test strip status information during the measurement process
- BloodGlucosePaperData: Blood glucose data

3) Sample Code

```
sdk.getBloodGlucoseData().listen((data) {
  if (data is BloodGlucoseSendData) {
    sendCmd(data.sendList);
  } else if (data is BloodGlucosePaperState) {
    bloodGlucose.value = data.message;
  } else if (data is BloodGlucosePaperData) {
    bloodGlucose.value = "blood glucose: ${data.data} ";
  }
});
```


2.3.8. Blood pressure

1) Function：

getBloodPressureData;

2) Callback data type：

- BloodPressureSendData:Data to be sent to the device
- BloodPressureProcess: Display the process of blood pressure measurement
- BloodPressureResult: Blood pressure data

  ps: systolic pressure

  pd: diastolic pressure

  hr: heart rate

3) Sample Code：

```
sdk.getBloodPressureData().listen((data) {
  if (data is BloodPressureSendData) {
    sendCmd(data.sendList);
  } else if (data is BloodPressureProcess) {
    debugPrint("BloodPressureProcess=${data.process}");
  } else if (data is BloodPressureResult) {
    bloodPressure.value =
        "SystolicBloodPressure:${data.ps}mmHg DiastolicBloodPressure:${data.pd}mmHg hr:${data.hr}times/minute";
    debugPrint(
        "BloodPressureResult ps=${data.ps} pd=${data.pd} hr=${data.hr}");
  }
}, onError: (Object error, StackTrace stackTrace) {
  if (error is AppException) {
    debugPrint("getBloodPressureData exception=${error.message}");
    bloodPressure.value = error.message;
  }
});
```

2.3.9. Battery

1) Function：

getBattery;

2) Callback data type：

- BatteryLevelData: Battery percentage data
- BatteryChargingStatus: Battery charging status

3) Sample Code：


```
sdk.getBattery().then((data) {
  if (data is BatteryLevelData) {
    battery.value = "battery: ${data.batteryLevel}%";
  } else if (data is BatteryChargingStatus) {
    battery.value = data.batteryCharging ? "charging" : "unCharging";
  }
});
```

2.3.10. Body temperature

1) Function：

getTemperature;

2) Callback data type：

TemperatureData: Body temperature data

3) Sample Code：

```
void startTemperature() {
  startDetect(Detection.BT);
  sdk.getTemperature().then((Hc03BaseMeasurementData data) {
    if (data is TemperatureData) {
      temperature.value = data.temperature;
    }
  }).catchError((error) {
    if (error is AppException) {
      debugPrint("getTemperature exception=${error.message}");
    }
  });
}
```