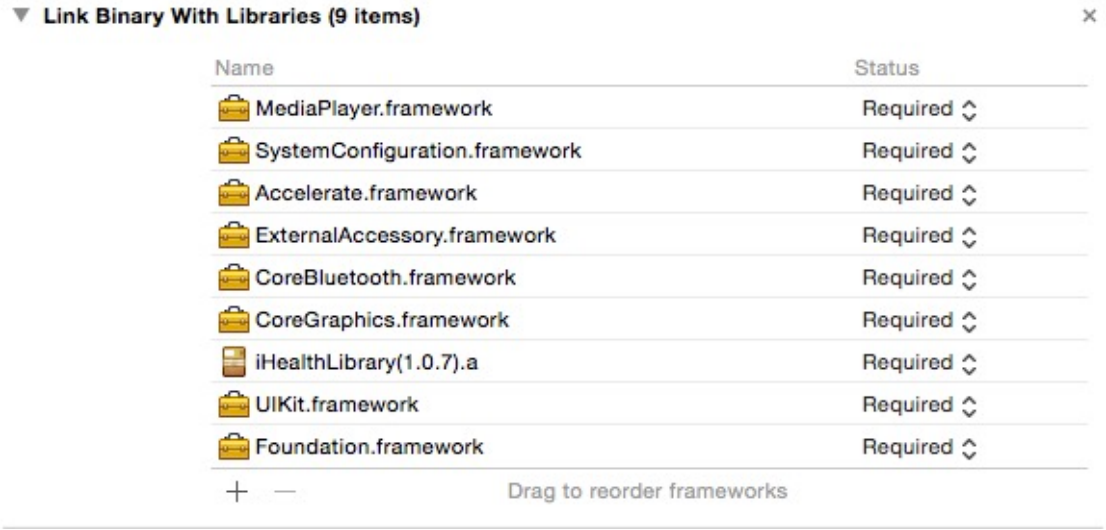
iHealthLabs EUROPE

Instruction Manual of BG SDK

1. Relevant file and Frameworks configuration

A. **BPSDK, including**: AudioBG1Communication.h，BG5.h，BG5Controller.h，BGMacroFile.h， BGHeader.h，iHealthLibrary(x.x.x).a, support iOS6.0+

B. **Frameworks**

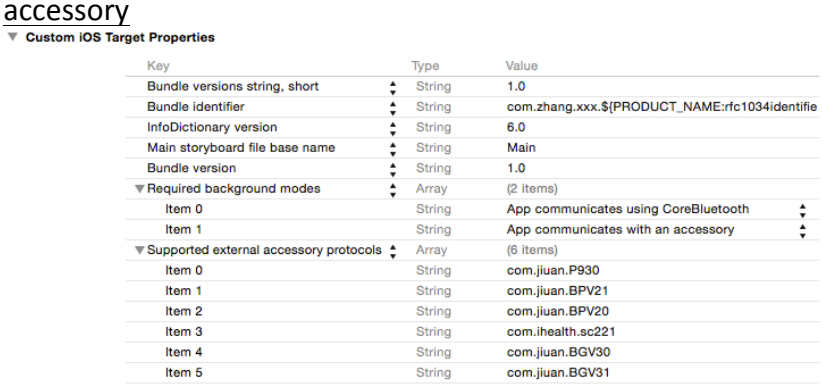


C. **Configuration**

Add new 'Item' in 'Info':

Add 2 new Item in '**Supported external accessory protocols**': com.jiuan.BGV30, com.jiuan.BGV31

Add 1 new Item in '**Required background modes**': App communicates with an accessory



2. Initialization

(1) Initialization for BG1 (connected BG via sound jack)

```
AudioBG1Communication
*audioBG1Communication=[AudioBG1Communication audioCommunicationObject];
```

- Using '**audioBG1Communication**' to call the communication module of the device

(2) Initialization for BG5 (wireless BG via Bluetooth)

```
BG5Controller *controller=[BG5Controller   shareIHBg5Controller];
BG5 *myBg5=[[controller getAllCurrentBG5Instace] objectAtIndex:0];
```

- Using '**controller getAllCurrentBG5Instace**' call communication module of the devices

(3) Resolve code

```
-(NSDictionary *)codeStripStrAnalysis:(NSString *)encodeString
```

Import parameters:

encodeString, code information by scanning the QR code.

Return parameter:

bottleID, bottleID,
DueDate, expired date.
StripNum, the number of strips

3. Interface Method:

BG1:

When the app detected the glucose meter, the notification with the name of BG1ConnectNoti **will be sent. After the device disconnected, the notification with the name of** BG1DisConnectNoti **will be received.**

(1) Establish measurement connection

```
-(void)commandCreateConnectWithUserID:(NSString*)userID
clientID:(NSString *)clientID clientSecret:(NSString *)clientSecret
Authentication:(DisposeAuthenticationBlock)disposeAuthenticationBlock
DisposeDiscoverBGBlock:(DisposeDiscoverBGBlock)disposeDiscoverBGBlock
DisposeBGIDPSBlock:(DisposeBGIDPSBlock)disposeBGIDPSBlock
DisposeConnectBGBlock:(DisposeConnectBGBlock)disposeConnectBGBlock
DisposeBGErrorBlock:(DisposeBGErrorBlock)disposeBGErrorBlock;
```

The authentication will be started by calling this method after the BG meter got connected.

Import parameters:

userID, as the only user label, is indicated by form of email address.

clientID and clientSecret, as the only user label, will be achieved after the register of SDK application. Please contact lvjincan@ihealthlabs.com.cn for the registration.

Return parameter:

disposeAuthenticationBlock is the return results after

the verification of **userID,clientID,clientSecret.**

Results:

a) **UserAuthen_RegisterSuccess,** new register successes.
b) **UserAuthen_LoginSuccess,** user logs in successfully.
c) **UserAuthen_CombinedSuccess,** user has been recognised as iHealth user, the measurement via SDK could be activated, the result data belongs to the user.
d) **UserAuthen_TrySuccess,** network error, the measurement is only for testing, SDK is not fully functional.
e) **UserAuthen_InvalidateUserInfo,** the verification of userID/clientID/clientSecret failed.
f) **UserAuthen_SDKInvalidateRight,** the application has not been authorised.
g) **UserAuthen_UserInvalidateRight,** the user has not been authorised.
h) **UserAuthen_InternetError,** network error, verification failed.

PS: 1. the measurement via SDK is functional in the case from a) to d).
2. the measurement via SDK will be determined in the case from e) to h), please contact iHealth support team, lvjincan@ihealthlabs.com.cn
3. "iHealth Disclaimer" will pop up and need to be proved by the user when SDK is activated for the first time.
4. if iHealth SDK has been using without internet, there is only 10-day try out because the SDK can not be certified.

disposeDiscoverBGBlock, blood glucose meter plugged in.
disposeBGIDPSBlock, to get the IDPS of the meter, this will be operated for the first time when the app talks to the meter.
disposeConnectBGBlock, the connection of the BG meter is regular , the measurement could be processed.
disposeBGErrorBlock, the connection is error, please refer to the error codes in the appendix

(2) Measurement establishing

```
-(void)commandCreateBGtestWithCode:(NSString*)codeStrips
DisposeBGSendCodeBlock:(DisposeBGSendCodeBlock)disposeBGSendCodeBlock
DisposeBGStripInBlock:(DisposeBGStripInBlock)disposeBGStripInBlock
DisposeBGBloodBlock:(DisposeBGBloodBlock)disposeBGBloodBlock
DisposeBGResultBlock:(DisposeBGResultBlock)disposeBGResultBlock
DisposeBGStripOutBlock:(DisposeBGStripOutBlock)disposeBGStripOutBlock
DisposeBGErrorBlock:(DisposeBGErrorBlock)disposeBGErrorBlock;
```

Import parameters:

codeStrips, get the QR code from the stripe bottle by app scanning.

Return parameter:

disposeBGSendCodeBlock, if the QR code is accepted, yes means accepted, no means deny.

disposeBGStripInBlock, yes means the strips slide into the BG meter.
disposeBGBloodBlock, yes means the blood drop has beed sensed from the strip.
disposeBGResultBlock, returns the measurement by the unit of mg/dL, range from 20-600.
disposeBGStripOutBlock, yes means the strip has been pulled out.
disposeBGErrorBlock , any errors during the measurement, pleaser refer to the error code list.

BG5:

BG5Controller *controller=[BG5Controller shareIHBg5Controller] needs to be called when the BG5 has been triggered. **When the app detected the glucose meter, the notification with the name of** BG5ConnectNoti **will be sent. After the device disconnected, the notification with the name of** BG5DisConnectNoti **will be received.**

(1) Establish measurement connection

```
-(void)commandInitBGSetUnit:(BGUnit )unitState
BGUserID:(NSString*)userID clientID:(NSString *)clientID
clientSecret:(NSString *)clientSecret
Authentication:(DisposeAuthenticationBlock)disposeAuthenticationBl
ock DisposeBGBottleID:(DisposeBGBottleID)disposeBGBottleID
DisposeBGErrorBlock:(DisposeBGErrorBlock)disposeBGErrorBlock;
```

Import parameters:

userID, as the only user label, is indicated by form of email address.
clientID and clientSecret, as the only user label, will be achieved after the register of SDK application. Please contact louie@ihealthlabs.com for the registration.
unitstate, BGUnit_mmolPL stands for mmol/L, **BGUnit_mgPmL** stands for mg/dL,

Return parameter:

disposeAuthenticationBlock is the return results after the verification of **userID,clientID,clientSecret.**
disposeBGBottleID, returns the ID which is stored in the BG meter, to verify if the strip has been used is from the same bottle of the registered one. if not, the app will notify the user need scan the new bottle, if yes, the app will get the number of left strips and expire date.
disposeBGErrorBlock, error code, please refer to the

error list.

(2) History records

```
-
(void)commandTransferMemorryData:(DisposeBGDataCount)disposeBGDataCo
unt DisposeBGHistoryData:(DisposeBGHistoryData)disposeBGHistoryData
DisposeBGErrorBlock:(DisposeBGErrorBlock)disposeBGErrorBlock;
```

Return parameter:

disposeBGDataCount, the number of the records in the meter memory.

disposeBGHistoryData , record detail, result means result, date means the measurement time.

disposeBGErrorBlock, error codes.

(3) Delete history records

```
-
(void)commandDeleteMemorryData:(DisposeBGDeleteData)DisposeBGDeleteD
ata;
```

Return parameter:

DisposeBGDeleteData, yes means deleted.

(4) Read the information of the code from the BG meter

```
-(void)commandReadBGCodeDic:(DisposeBGCodeDic)disposeBGCodeDic
DisposeBGErrorBlock:(DisposeBGErrorBlock)disposeBGErrorBlock;
```

Return parameter:

disposeBGCodeDic, returns the information of the code, **bottleID** means the strip ID, **Strips** means the number of strips which has been used, **Date** means expired date.
disposeBGErrorBlock, error codes.

(5) Send code

```
-(void)commandSendBGCodeString:(NSString*)encodeString
bottleID:(NSNumber *)bottleID validDate:(NSDate*)date
remainNum:(NSNumber*)num
DisposeBGSendCodeBlock:(DisposeBGSendCodeBlock)disposeBGSendCodeBloc
k DisposeBGStartModel:(DisposeBGStartModel)disposeBGStartModel
DisposeBGErrorBlock:(DisposeBGErrorBlock)disposeBGErrorBlock;
```

Import parameters:

encodeString, gets the code information by scanning the QR code.
bottleID, bottleID,

date, expired date.
num, the number of strips which is last. num = 0 will determine the measurement.

Return parameter:

disposeBGSendCodeBlock, yes means code has been sent regularly.
disposeBGStartModel, the boot mode of the BG meter, **BGOpenMode_Strip** means booting the meter by sliding the strip, **BGOpenMode_Hand** means booting the meter by pressing the on/off button. **interface (6)** will be called by the first mode, **interface (7)** will be called by the second mode.
disposeBGErrorBlock, error codes.

(6) Strip-Sliding booting mode

```
-
(void)commandCreateBGtestStripInBlock:(DisposeBGStripInBlock)dispose
BGStripInBlock
DisposeBGBloodBlock:(DisposeBGBloodBlock)disposeBGBloodBlock
DisposeBGResultBlock:(DisposeBGResultBlock)disposeBGResultBlock
DisposeBGTestModelBlock:(DisposeBGTestModelBlock)disposeBGTestModelB
lock DisposeBGErrorBlock:(DisposeBGErrorBlock)disposeBGErrorBlock
```

Return parameter:

disposeBGStripInBlock, yes means strip slides in.
disposeBGBloodBlock, yes means the blood drop has beed sensed from the strip
disposeBGResultBlock, returns the measurement by the unit of mg/dL, range from 20-600.
disposeBGTestModelBlock, measurement mode, **BGMeasureMode_Blood** means blood measurement mode, **BGMeasureMode_NoBlood** means control solution measurement mode.
disposeBGErrorBlock, error codes.

(7) Button-pressing booting mode

```
-(void)commandCreateBGtestModel:(BGMeasureMode)testMode
DisposeBGStripInBlock:(DisposeBGStripInBlock)disposeBGStripInBlock
DisposeBGBloodBlock:(DisposeBGBloodBlock)disposeBGBloodBlock
DisposeBGResultBlock:(DisposeBGResultBlock)disposeBGResultBlock
DisposeBGTestModelBlock:(DisposeBGTestModelBlock)disposeBGTestModelB
lock  DisposeBGErrorBlock:(DisposeBGErrorBlock)disposeBGErrorBlock;
```

Import parameters:

testMode measurement mode, **BGMeasureMode_Blood** means blood measurement mode, **BGMeasureMode_NoBlood** means control solution measurement mode.

Return parameter:

disposeBGStripInBlock, yes means strip slides in.
disposeBGBloodBlock, yes means the blood drop has beed sensed from the strip
disposeBGResultBlock, returns the measurement by the unit of mg/dL, range from 20-600.
disposeBGTestModelBlock, measurement mode, **BGMeasureMode_Blood** means blood measurement mode, **BGMeasureMode_NoBlood** means control solution measurement mode.
disposeBGErrorBlock, error codes.

(8) Resolve code

```
-(NSDictionary *)codeStripStrAnalysis:(NSString *)encodeString
```

Import parameters:

encodeString, code information by scanning the QR code.

Return parameter:

bottleID, bottleID,
DueDate, expired date.
StripNum, the number of strips

4. Error Code

00: Battery is low.
01: Glucose test result is out of the measurement range.
02: Unknown interference detected, please repeat the test.
03: Strip is used or unknown moisture detected, discard the test strip and repeat the test with a new strip.
04: Reading transmission error. Repeat the test with a new test strip. If the problem persists, contact iHealth customer service for assistance.
05\06: The environmental temperature is beyond normal range, place the meter at room temperature for at least 30 minutes, then repeat the test.
07: Test strip coding error.
08: Communication error, press“START” or rescan the code to repeat the test.
09: Strip removed in the middle of reading, repeat the test with a new strip.
10: Insert a new test strip and repeat the test.
11: Cannot write to SN or KEY.
12: Please set time.
13: 0 test strips remaining.
14: Test strip expired.
15: Unplug the charging cable before testing.
100: BG meter disconnected.
101: BG meter is in sleeping mode, needs repair.
111: user verification failed.

5. Demo