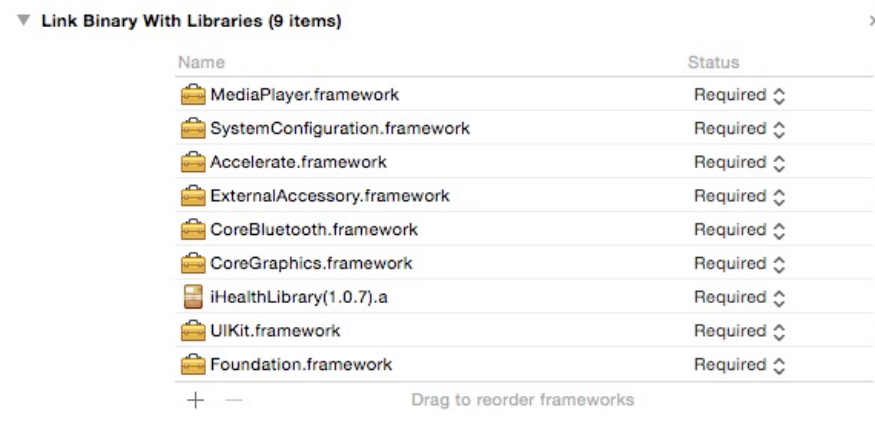


iOS AM SDK Documentation

1. Relevant files and frameworks

- (1) Import the following AM SDK files: AMHeader.h、AMMacroFile.h、AM3.h、AM3Controller.h、AM3S.h、AM3SController.h、AM4.h、AM4Controller.h、User.h 、iHealthLibrary(x.x.x).a, supports iOS 6.0 and above.

- (2) Frameworks



- (3) Configuration

Add an "item" in "Info"

Add two new items in "Required background modes", App communicates with an accessory 、App communicates using CoreBluetooth

▼ Custom iOS Target Properties

Key	Type	Value
Bundle versions string, short	String	1.0
Bundle identifier	String	com.zhang.xxx.\${PRODUCT_NAME:rfc1034identifie
InfoDictionary version	String	6.0
Main storyboard file base name	String	Main
Bundle version	String	1.0
▼ Required background modes	Array	(2 items)
Item 0	String	App communicates using CoreBluetooth
Item 1	String	App communicates with an accessory
▼ Supported external accessory protocols	Array	(6 items)
Item 0	String	com.jiuan.P930
Item 1	String	com.jiuan.BPV21
Item 2	String	com.jiuan.BPV20
Item 3	String	com.ihealth.sc221
Item 4	String	com.jiuan.BGV30
Item 5	String	com.jiuan.BGV31

2. Operation Procedure

(1) AM3 instructions

Register device AM3: `AM3ConnectNoti`;

Initialize AM3 controller class:

```
AM3Controller *controller = [AM3Controller sharedInstance];
```

Access control class instance after receiving `AM3ConnectNoti`:

```
NSArray *amDeviceArray = [controller getAllCurrentAM3Instace];
AM3 *amInstance = [amDeviceArray objectAtIndex:i];
Use amInstance to call AM3 related communication methods.
```

If already connected to the correct AM3, can stop connections to any other AM3's with the following API:

```
[controller commandCanConnectAM:], True: Can connect to AM3's. False: Stop connecting to other AM3's.
```

(2) AM3S instructions

Register device AM3S: `AM3SConnectNoti`;

Initialize AM3S controller class:

```
AM3SController *controller = [AM3SController sharedInstance];
Access control class instance after receiving AM3SConnectNoti:
NSArray *amDeviceArray = [controller getAllCurrentAM3Instace];
AM3S *amInstance = [amDeviceArray objectAtIndex:i];
Use amInstance to call AM3s related communication methods.
a. To specify a specific AM3s device to improve the efficiency of the connection,
    use the following API:
[controller commandSetYourDeviceID:],Specify the AM3s unique ID (MAC Address).

b. If already connected to the correct AM3, can stop connections to any other
    AM3's with the following API:
[controller commandCanConnectOtherDevice:], True: Can connect to AM3s's. False:
Stop connecting to other AM3s's.
```

(3) AM4 instructions

Register device AM4: `AM4ConnectNoti`;

Initialize AM4 controller class:

```
AM4Controller *controller = [AM4Controller sharedInstance];
Access control class instance after receiving AM4ConnectNoti:
NSArray *amDeviceArray = [controller getAllCurrentAM4Instace];
AM4 *amInstance = [amDeviceArray objectAtIndex:i];
Use amInstance to call AM4 related communication methods.
c. To specify a specific AM4 device to improve the efficiency of the connection,
    use the following API:
[controller commandSetYourDeviceID:],Specify the AM4 unique ID (MAC Address).

d. If already connected to the correct AM4, can stop connections to any other
    AM3's with the following API:
```

[controller commandCanConnectOtherDevice:], True: Can connect to AM4's. False:
Stop connecting to other AM4's.

3. AM3 Interface Instructions

(1) Establish memory and measurement connection

Only after verification through this interface can we move onto using other API's.

```
-(void)commandCreateUserManageConnectWithUser:(User *)tempUser  
Authentication:(BlockUserAuthentication)disposeAuthenticationBlock  
currentUserSerialNub:(CurrentSerialNub)serialNub  
amUser:(DisposeAM3AskUserID)disposeAskUserID  
binedAMSerialNub:(DisposeBinedAMSerialNub)binedSerialnub  
currentSerialNub:(DisposeCurrentSerialNub)currentSerialNub  
DisposeErrorBlock:(DisposeAM3ErrorBlock)disposeErrorBlock;
```

Input Parameters:

tempUser, includes properties: clientID, clientSecret, userID。

userID, either email or mobile phone number (mobile phone number not yet supported).

ClientID and clientSecret, the only identification for users of the SDK, requires registration from iHealth administrator, please email: lvjincan@ihealthlabs.com.cn for more information

Return Parameters:

disposeAuthenticationBlock: The return parameters of "userid", 'clientID', and 'clientSecret' after verification。

The interpretation for the verification:

UserAuthen_RegisterSuccess: New-user registration succeeded.

UserAuthen_LoginSuccess: User login succeeded.

UserAuthen_CombinedSuccess: The user is an iHealth user as well, measurement via SDK has been activated, and the data from the measurement belongs to the user.

UserAuthen_TrySuccess: Testing without internet connection succeeded.

UserAuthen_InvalidateUserInfo: Userid/clientID/clientSecret verification failed.

UserAuthen_SDKInvalidateRight: SDK has not been authorized.

UserAuthen_UserInvalidateRight: User has not been authorized.

UserAuthen_InternetError: Internet error, verification failed.

The measurement via SDK will be operated in the case of 1-3, and will be terminated if any of 4-8 occurs. The interface needs to be re-called after analyzing the return parameters.

Notice: when a new user registers via SDK, an 'iHealth disclaimer' will pop up automatically, and will require the user to agree in order to continue. SDK applications require an Internet connection.

SerialNub: Uniquely identifies the user, the SDK requires this to be stored. This ID will be sent to the AM3 and will allow the AM3 to pair with only this user.

disposeAskUserID: The user ID that is stored on the AM3, 0 indicates that there is no user info

binSerialnub: The user's AM3's MAC Address

currentSerialNub: The connected user's MAC Address

disposeErrorBlock: Communication error codes, see section 6 AM3 error descriptions.

(2) Pair/bond users

For use when the AM3 returns a user ID of 0:

Note: Bonding a user account and AM3 requires an active internet connection.

```
-(void)commandsetAM3UserID:(NSNumber*)userID  
DisposeBlock:(DisposeAM3Block)disposeBlock  
DisposeErrorBlock:(DisposeAM3ErrorBlock)disposeErrorBlock;
```

Input parameters:

userID.

Return parameters:

disposeBlock : YES: Account bonding successful. NO: Failed

disposeErrorBlock: Communication error codes, see section 6 AM3 error descriptions.

(3) AM3 initialization

Must be called the first time to ensure that the AM3 has correct user information, goals, time, battery checks, etc.

```
-(void)commandSyncUserInfoWithUser:(User *)tempUser  
andGoal:(NSNumber*)goalNumber  
DisposeStateInfo:(DisposeAM3StateInfo)disposeStateInfo  
DisposeBattery:(DisposeAM3Battery)disposeBattery  
DisposeBlock:(DisposeAM3Block)disposeBlock  
DisposeErrorBlock:(DisposeAM3ErrorBlock)disposeErrorBlock;
```

Input parameters:

tempUser, User information, needs to include the following: birthday、height、weight、bmr、sex、lengthUnit.

birthday, NSDate.

height, (cm).

weight, (kg).

bmr, user basal metabolic rate.

sex, UserSex_Female or UserSex_Male.

lengthUnit, total distance, LengthUnit_Mile is imperial units,
LengthUnit_Kilometer for metric units.
goalNumber, User goal number of steps. Default is 10,000

Return parameters:

disposeStateInfo: AM status, State_wrist (AM3 being worn on the wrist),
State_waist (AM3 worn with belt clip).
disposeBattery: AM battery percentage, from 0~100.
disposeBlock: Initialization complete. True: Success, False: Failed.
disposeErrorBlock: see section 6 AM3 error descriptions.

(4) Restore factory settings

-(void)commandResetDeviceDisposeResultBlock:(DisposeAM3Block)disposeBlock
DisposeErrorBlock:(DisposeAM3ErrorBlock)disposeErrorBlock;

Return parameters:

disposeBlock: True: Success, False: Failed.
disposeErrorBlock: see section 6 AM3 error descriptions.

(5) AM3 Alarm

-(void)commandQueryAlarmInfo:(DisposeAM3TotoalAlarmData)disposeTotoalAlarmD
ata DisposeErrorBlock:(DisposeAM3ErrorBlock)disposeErrorBlock;

Return parameters:

disposeTotoalAlarmData : Alarm array contains up to 3 alarms, each one needs
the following parameters: AlarmId、Time、IsRepeat、Switch、(Sun、Mon、
Tue、Wed、Thu、Fri、Sat)
AlarmId: 1, 2, 3
Time: HH:mm。
IsRepeat: Repeat alarm, True: Repeat, False: Don't repeat.
Switch: Alarm on/off. True: On, False: Off
Sun、Mon、Tue、Wed、Thu、Fri、Sat: True.
disposeErrorBlock: see section 6 AM3 error descriptions.

(6) Set Alarm

-(void)commandSetAlarmWithAlarmDictionary:(NSDictionary *)alarmDic
DisposeResultBlock:(DisposeAM3Block)disposeBlock
DisposeErrorBlock:(DisposeAM3ErrorBlock)disposeErrorBlock

Input parameters:

alarmDic: Alarm information, include parameters: AlarmId、Time、IsRepeat、
Switch、(Sun、Mon、Tue、Wed、Thu、Fri、Sat,

Return parameters:

disposeBlock: True: Alarm set successfully, False: Failed。
disposeErrorBlock: see section 6 AM3 error descriptions.

(7) Delete alarm

-(void)commandDeleteAlarmVialID:(NSNumber *)alarmID
DisposeResultBlock:(DisposeAM3Block)disposeBlock
DisposeErrorBlock:(DisposeAM3ErrorBlock)disposeErrorBlock;

Input parameter:

alarmID: 1, 2, 3

Return parameters:

disposeBlock: True: Delete successful, False: Failed
disposeErrorBlock: see section 6 AM3 error descriptions.

(8) Query reminder

-(void)commandQueryReminder:(RemindAM3Info)remindInfo
DisposeErrorBlock:(DisposeAM3ErrorBlock)disposeErrorBlock;

Input parameters:

remindInfo: Array containing following parameters: **Time**、**Switch**。
Time: format HH:mm, time between reminders (HH*60+mm) minutes。
Switch: Reminder on/off, True: On, False: Off.
disposeErrorBlock: see section 6 AM3 error descriptions

(9) Set reminders

-(void)commandSetReminderwithReminderDictionary:(NSDictionary
*)reminderDic DisposeResultBlock:(DisposeAM3Block)disposeBlock
DisposeErrorBlock:(DisposeAM3ErrorBlock)disposeErrorBlock;

Input parameters:

reminderDic: Array containing collowing parameters: **Time**、**Switch**。

Return parameters:

disposeBlock: True: Success, False: Failed.
disposeErrorBlock: see section 6 AM3 error descriptions

(10) Upload AM3 data

Data type: 5 minutes of motion data, 5 minutes of sleep data, total number of steps for the day, and total calories. Also includes the number of steps for the 5 minutes of motion data, total calories for the current time, calories of the steps, and total calories. If calculations for every 5 minutes of motion data is required, you will need to calculate the difference between two records.

-(void)commandSyncAllAMData:(StartAM3Transmission)startAM3Transmissio
DisposeProgress:(DisposeAM3ProgressData)disposeAM3ProgressData
historyData:(AM3HistoryData)AM3historyData
FinishTransmission:(FinishAM3Transmission)finishAM3Transmission
startsleepdata:(StartSleepTransmission) startSleepTransmission

DisposeSleepProgress:(DisposeSleepProgressData)disposeSleepProgressData
 sleepHistoryData:(SleepHistoryData)sleepHistoryData
 FinishSleepTransmission:(FinishSleepTransmission)finishSleepTransmission
 CurrentActiveInfo:(DisposeAM3QueryCurrentActiveInfo)
 disposeQueryCurrentActiveInfo
 DisposeErrorBlock:(DisposeAM3ErrorBlock)disposeErrorBlock
 AM3IsOnTransmission:(AM3IsOnTransmission)am3isOnTransmission
 SleepIsOnTransmission:(SleepIsOnTransmission)sleepisOnTransmission;

Return parameters:

startAM3Transmissio: Start uploading motion data, including parameters:

StartActiveHistoryDate、StepSize、StartActiveHistoryTotoalNum。

StartActiveHistoryDate: Start date, yyyy-MM-dd.

StepSize: Length of each step, cm。

StartActiveHistoryTotoalNum: Number of records

disposeAM3ProgressData: AM data upload percentage, 0.0~1.0.

AM3historyData: Workout data, including the following parameters: AMDate、AMCalorie、AMStepNum。

AMDate: Workout time, NSDate.

AMCalorie: Current time total calories。

AMStepNum: Total number of steps。

finishAM3Transmission: Upload complete.

startSleepTransmission: Start uploading sleep data, including parameters:

SleepActiveHistoryDate、StartActiveHistoryTotoalNum。

SleepActiveHistoryDate: Sleep start time, yyyy-MM-dd HH:mm:ss.

StartActiveHistoryTotoalNum: Number of records

disposeSleepProgressData: AM sleep data upload percent 0.0~1.0。

sleepHistoryData: Sleep data, including the following parameters: AMDate、SleepData。

AMDate: Sleep time, NSDate.

SleepData: Sleep grade, 0: awake, 1: light sleep, 2: deep sleep.

finishSleepTransmission: Upload complete.

disposeQueryCurrentActiveInfo: Total calories and steps for today, including parameters: Step、Calories。

Step: Number of steps taken today。

Calories: Number of calories burned today。

disposeErrorBlock: see section 6 AM3 error descriptions

am3isOnTransmission: Invalidate.

sleepisOnTransmission: Invalidate.

(11) Set AM3 status

-(void)commandSetState:(ActiveState)activeState

DisposeBlock:(DisposeAM3Block)disposeBlock

DisposeErrorBlock:(DisposeAM3ErrorBlock)disposeErrorBlock;

Input parameters:

activeState: Sleep_State: sleep mode, Active_State: active, Fly_State: airplane mode, Drive_State: driving mode。

Return parameters:

disposeBlock: True: Set successfully, False: Failed.

disposeErrorBlock: see section 6 AM3 error descriptions

(12) Disconnect AM3 connection

-(void)commandDisconnectDisposeBlock:(DisposeAM3Block)disposeBlock

DisposeErrorBlock:(DisposeAM3ErrorBlock)disposeErrorBlock;

Return parameters:

disposeBlock: True: Success, False: Failed.

disposeErrorBlock: see section 6 AM3 error descriptions

4、AM3S Interface Instructions

AM3S functionalities are similar to the previous section for AM3, this section will only describe the differences

(1) Sending a random number

This API sends a random number to the AM3. Only when the random number matches the number displayed on the AM3s screen can the device be bound to the device.

-(void)commandSetRandomString:(DisposeRandomNumberSetting)disposeBlock

DisposeErrorBlock:(DisposeAM3SErrorBlock)disposeErrorBlock;

Return parameters:

disposeBlock, True: Sent successfully, False: Failed。 Random number is six digits, ranging from 0 – 999999. AM3s will receive the random number and display on screen. The user will have to enter it into the app.

disposeErrorBlock: see section 6 AM3S error descriptions

(2) Binding AM3s to user

Note: Account binding requires an active internet connection

-(void)commandSetAM3SUserID:(NSNumber*)userID withRandom:(NSString*)tempRandom DisposeBlock:(DisposeAM3SBlock)disposeBlock

DisposeErrorBlock:(DisposeAM3SErrorBlock)disposeErrorBlock;

Input parameters:

userID

tempRandom, the 6 random numbers displayed on the AM3s。

Return parameters:

disposeBlock : YES: Binding successful, NO: Failed.

disposeErrorBlock: see section 6 AM3S error descriptions

(3) Set AM3s state

-(void)commandSetState:(AM3SActiveState)activeState
DisposeBlock:(DisposeAM3SBlock)disposeBlock
DisposeErrorBlock:(DisposeAM3SErrorBlock)disposeErrorBlock;

Input parameters:

activeState, AM3SFly_State: Airplane mode

Return parameters:

disposeBlock : YES: Successfully set, NO: Failed.
disposeErrorBlock: see section 6 AM3S error descriptions

(4) Upload AM3s report data

-(void)commandSetSyncSportCount:(DisposeSyncSportCount)disposeSyncSportCount
DisposeMeasureData:(DisposeMeasureData)disposeMeasureData
disposeFinishMeasure:(DisposeFinishMeasure)disposeFinishMeasure
DisposeErrorBlock:(DisposeAM3SErrorBlock)disposeErrorBlock;

Return parameters:

disposeSyncSportCount: Total report number.
DisposeMeasureData: Report data, including parameters:
ReportStage_Swimming、ReportStage_Work_out、
ReportStage_Sleep_summary、ReportStage_Activeminute. Currently only
supports ReportStage_Work_out、ReportStage_Sleep_summar。
Workout contains properties: ReportState、Work_outMeasureDate、
Work_outTimeNumber、Work_outStepNumber、Work_outLengthNumber、
Work_outCalories。
ReportState: ReportStage_Work_out。
Work_outMeasureDate: Start time。
Work_outTimeNumber: Length of workout (mins)。
Work_outStepNumber: Workout number of steps。
Work_outLengthNumber: Workout distance (km)。
Work_outCalories: Workout calories burned。

Sleep contains properties: ReportState、Sleep_summaryMeasureDate、
Sleep_summarySleepTime、Sleep_summarysleepEfficiency、
Sleep_summarysleepAddMinute。
ReportState: ReportStage_Sleep_summary。
Sleep_summaryMeasureDate: Sleep start time。
Sleep_summarySleepTime: Sleep duration (mins)。
Sleep_summarysleepEfficiency: Sleep efficiency percentage, range is 0-100.
Sleep_summarysleepAddMinute: Correct sleep duration length. Change the
length of time from before falling asleep to add onto the time awake.
disposeErrorBlock: see section 6 AM3S error descriptions

5、AM4 Interface Instructions

AM4 functionalities are similar to the previous section for AM3, this section will only describe the differences

(1) Sending a random number

This API sends a random number to the AM3. Only when the random number matches the number displayed on the AM4 screen can the device be bound to the device.

-(void)commandAM4SetRandomBlock:(DisposeAM4SetRandomBlock)disposeSetRandom disposeErrorBlock:(DisposeAM4ErrorBlock)disposeErrorBlock;

Return parameters:

disposeSetRandom, True: Sent successfully, False: Failed。 Random number is six digits, ranging from 0 – 999999. AM4 will receive the random number and display on screen. The user will have to enter it into the app.

disposeErrorBlock: see section 7 AM4 error descriptions

(2) Binding AM4 to user

Note: Account binding requires an active internet connection

-(void)commandSetAM4UserID:(NSNumber*)userID withRandom:(NSString*)tempRandom DisposeBlock:(DisposeAM4SetUserIDBlock)disposeBlock DisposeErrorBlock:(DisposeAM4ErrorBlock)disposeErrorBlock;

Input parameters:

userID

tempRandom, the 6 random numbers displayed on the AM4。

Return parameters:

disposeBlock : YES: Binding successful, NO: Failed.

disposeErrorBlock: see section 7 AM4 error descriptions

(3) Set AM4 state

-(void)commandAM4SetState:(AM4ActiveState)activeState disposeBlock:(DisposeAM4SetStateBlock)disposeSetState disposeErrorBlock:(DisposeAM4ErrorBlock)disposeErrorBlock;

Input parameters:

activeState, AM4Fly_State

Return parameters:

disposeSetState: YES: Successfully set, NO: Failed.

disposeErrorBlock: see section 7AM4 error descriptions

(4) Set Swimming

-(void)commandAM4SetSwimmingState:(BOOL)swimmingIsOpen swimmingPoolLength:(NSNumber*)swimmingPoolLength NOswimmingTime:(NSDate*)noSwimmingDate unit:(AM4SwimmingUnit)unit resultBlock:(DisposeAM4SettingSwimmingBlock)disposeSetSwimming disposeErrorBlock:(DisposeAM4ErrorBlock)disposeErrorBlock;

Input parameters:

swimmingIsOpen YES:open swimming function NO:close swimming function default:no

swimmingPoolLength swimming Pool Length

noSwimmingDate automatic drop out swim duration

unit swim unit

Return parameters:

disposeSetSwimming: YES: Successfully set, NO: Failed.

disposeErrorBlock: see section 7AM4 error descriptions

(5) Upload AM4 report data

```
-(void)commandAM4SetSyncsportCount:(DisposeAM4SyncSportCountBlock)disposeSyncSportCount disposeMeasureData:(DisposeAM4MeasureDataBlock)disposeMeasureData disposeFinishMeasure:(DisposeAM4WorkoutFinishBlock)disposeFinishMeasure disposeErrorBlock:(DisposeAM4ErrorBlock)disposeErrorBlock;
```

Return parameters:

disposeSyncSportCount: Total report number.

disposeMeasureData: Report data, including parameters:

ReportStage_Swimming、ReportStage_Work_out、

ReportStage_Sleep_summary、ReportStage_Activeminute. Currently only

supports ReportStage_Work_out、ReportStage_Sleep_summar、

ReportStage_Swimming。

Workout contains properties: AM4ReportState、AM4Work_outMeasureDate、

AM4Work_outTimeNumber、AM4Work_outStepNumber、

AM4Work_outLengthNumber、AM4Work_outCalories。

AM4ReportState: AM4ReportStage_Work_out。

AM4Work_outMeasureDate: Start time。

AM4Work_outTimeNumber: Length of workout (mins)。

AM4Work_outStepNumber: Workout number of steps。

AM4Work_outLengthNumber: Workout distance (km)。

AM4Work_outCalories: Workout calories burned。

Sleep contains properties: AM4ReportState、AM4Sleep_summaryMeasureDate、

AM4Sleep_summarySleepTime、AM4Sleep_summarysleepEfficiency、

AM4Sleep_summarysleepAddMinute。

AM4ReportState : AM4ReportStage_Sleep_summary。

AM4Sleep_summaryMeasureDate: Sleep start time.

AM4Sleep_summarySleepTime: Sleep duration (mins)。

AM4Sleep_summarysleepEfficiency: Sleep efficiency percentage, range is

0-100.

AM4Sleep_summarysleepAddMinute: Correct sleep duration length. Change the length of time from before falling asleep to add onto the time awake.

disposeErrorBlock: see section 7AM4 error descriptions

Swimming contains properties: AM4ReportState、AM4SwimmingMeasureDate、

AM4SwimmingTimeNumber、AM4SwimmingTimes、AM4Swimmingcalories、AM4SwimmingAct、

AM4SwimmingCircleCount、AM4SwimmingPoollength。

AM4ReportState: AM4ReportStage_Swimming

AM4SwimmingMeasureDate: finish Time

AM4SwimmingTimeNumber: Swimming Processed

AM4SwimmingTimes: Swimming Times
AM4Swimmingcalories: swimming calories burned
AM4SwimmingAct: Swimming posture
AM4SwimmingCircleCount: Swimming Circle Count
AM4SwimmingPoollength: Swimming Pool length

6. Additional information

Device connection info: [AM3ConnectNoti](#)、[AM3SConnectNoti](#)、[AM4ConnectNoti](#)

Device disconnection info: [AM3DisConnectNoti](#)、[AM3SDisConnectNoti](#)、[AM4DisConnectNoti](#)

When connecting multiple devices, distinguish between them via the serialNumber attribute.

7. Error codes

(1)AM3:

```
typedef enum{  
    AMErrorOverTime = 0,//Communication error  
    AM_Reset_Device_Faild,//Reset failed  
    AMErrorDisconnect,//AM disconnect  
    AMErrorUserInvalidate//invalidate user info  
}AMErrorID;
```

(2)AM3S:

```
typedef enum{  
    AM3SErrorOverTime = 0, //Communication error  
    AM3S_Reset_Device_Faild,//Reset failed  
    AM3SErrorDisconnect, //AM disconnect  
    AM3SErrorUserInvalidate//invalidate user info  
}AM3SErrorID;
```

(3)AM4:

```
typedef enum{  
    AM4ErrorOverTime = 0,  
    AM4_Reset_Device_Faild,  
    AM4ErrorDisconnect,  
    AM4ErrorUserInvalidate  
}AM4ErrorID;
```

7. demo