

# README

## How to use iot-sdk

### 1. 简介

本 sdk 提供一个基于 Linux和STM32F411RE, 通过与 Wi-Fi 模块进行 AT 指令串口通信, 实现与阿里云 ILOP 和 sds 以及中国移动杭研平台 直连的 IoT 物联网典型开发应用示例源代码,master分支是基于linux平台, NUCLEOF411RE分支是基于IAR和GCC平台。

### 2. 目录结构

```
1 |---example
2 |   |---ilopicaapp /*ilop平台 ica标准数据格式的demo*/
3 |   |---iloprawapp /*ilop平台 自定义数据格式的demo*/
4 |   |---sdsapp /*sds平台的demo*/
5 |   |---qlinkapp /*杭研平台的demo*/
6 |   |---test /*测试demo*/
7 |
8 |---platform
9 |   |---linux /*linux平台相关的接口实现*/
10 |   |---mx_hal.h
11 |
12 |---src
13 |   |---alicloud_ilop /*ilop平台的实现*/
14 |   |---alicloud_sds /*sds平台的实现*/
15 |   |---helper /*at指令解析内核*/
```

### 3. Linux 平台开发

#### 3.1 准备

- 串口接入PC, 并查询设备, 命令: `ls -l /dev/ttyUSB*`

```
1 parallels@parallels-vm:~/iot-sdk$ ls -l /dev/ttyUSB*
2 crw-rw---- 1 root dialout 188, 0 Sep  3 16:15 /dev/ttyUSB0
```

#### 3.2 编译

iot-sdk 提供四个demo, 编译demo的命令格式 `make <demo>`

- 清除上次的make命令所产生的build文件, 命令: `make clean`

```
1 parallels@parallels-vm:~/iot-sdk$ make clean
2 cleaning...
3 done
```

- 编译测试demo, 命令: `make test`

```
1 parallels@parallels-vm:~/iot-sdk$ make test
```

```

2 mkdir build
3 compiling platform/linux/mx_serial.c
4 compiling platform/linux/mx_stdio.c
5 compiling platform/linux/mx_tick.c
6 compiling platform/linux/mx_uart.c
7 compiling src/helper/cli/mx_cli.c
8 compiling src/helper/emhost/ATCmdParser/ATCmdParser.c
9 compiling src/helper/emhost/emh_alisds.c
10 compiling src/helper/emhost/emh_arg.c
11 compiling src/helper/emhost/emh_iloc.c
12 compiling src/helper/emhost/emh_module.c
13 compiling src/helper/emhost/emh_wlan.c
14 compiling src/helper/mx_utils/mx_ringbuffer.c
15 compiling examples/test/test.c
16 generating build/iot-test.elf
17      text      data      bss      dec      hex filename
18  16294      928     3680     20902     51a6 build/iot-test.elf

```

- 编译完成后，在build目录下生成 `iot-test.elf` 文件

### 3.3 运行

- 运行build目录下的elf文件，命令： `./build/iot-test.elf <dev>`

```

1 parallels@parallels-vm:~/iot-sdk$ sudo ./build/iot-test.elf /dev/ttyUSB0
2 [sudo] password for parallels:
3 dev[/dev/ttyUSB0]
4 open at uart succeed
5 [APP: test.c: 19] FW version: ilop_AT_v2.1.4
6 [APP: test.c: 20] System tick: 757

```

## 4. IoT示例

### 4.1 阿里 ilop 平台的标准数据格式demo

### 4.2 阿里 ilop 平台的透传demo

### 4.3 杭研 qlink 平台的demo

#### 4.3.1 使用流程

- (1) 获取客户在杭研申请账号密码
- (2) 商务洽谈后，提交资料，完成后台产品创建，后台地址：<https://open.home.komect.com/partner/>
- (3) 搭建虚拟网关开发环境（详见examples/qlink中materials文件夹）
- (4) 进行本地能网关环境下验证(使用虚拟网关开发调试)
- (5) 杭研现场验收
- (6) 项目入库
- (7) 厂家提交配网相关资料，杭研配置成XML文件集成到H5开发的和家亲APP(此步骤或许可以提前，需要厂商和杭研沟通)
- (8) 使用materials/下的和家亲APP测试(开发环境获取到的mqtt服务器地址以.199结尾,测试环境以.207结尾，生产环境以205结尾)

## (9) 杭研云网关验证

### 4.3.2 使用方法

#### (1) 庆科wifi模块烧录杭研qlink最新[AT固件](#)

固件烧录具体方法可以参考resource/common/烧录路径下相关文档。

#### (2) 修改产品信息

新建产品后，可以获得product\_token, andlink\_token, product\_id, 豪恩报警盒子产品信息如下：

产品信息

产品名称

T1报警盒子

产品分类

产品品牌

豪恩

连接类型

产品ID

30531

产品验证码  
(productToken)

PXmsE6kQmxEgc0e3

平台验证码  
(andlinkToken)

ZxMo0zJvvZMWP3Az

创建时间

2018-01-22 10:54:50

描述

它作为信息枢纽，不但实现无线传感器的组网以及与上行网络的通讯，而且也可以实现一些报警应用场景。

```
1 初始化产品参数
2 const emh_qlink_config_t config = {
3     .product_info = {
4         .product_id    = "30531",
5         .product_token = "PXmsE6kQmxEgc0e3",
6         .andlink_token = "ZxMo0zJvvZMWP3Az",
7         .format = EMH_ARG_QLINK_FORMAT_JSON,
8     },
9     .version_info = {
10         .firmware_version = "1.4.3",
11         .software_version = "1.0.1",
12     }
13 };
14
```

#### (3) 编译运行demo

- make clean

```
1 wumingju@ubuntu:/mnt/hgfs/Work/project/AT_demo/at_host_qlink$ make clean
2 cleaning...
```

3 done

- make qlinkapp

```
1 mkdir build
2 compiling platform/linux/mx_serial.c
3 compiling platform/linux/mx_stdio.c
4 compiling platform/linux/mx_tick.c
5 compiling platform/linux/mx_uart.c
6 compiling src/helper/cli/mx_cli.c
7 compiling src/helper/emhost/ATCmdParser/ATCmdParser.c
8 compiling src/helper/emhost/emh_alisds.c
9 compiling src/helper/emhost/emh_arg.c
10 compiling src/helper/emhost/emh_ilop.c
11 compiling src/helper/emhost/emh_qlink.c
12 generating build/iot-qlinkapp.elf
13      text      data      bss      dec      hex filename
14  22826      1168      3696      27690      6c2a build/iot-qlinkapp.elf
```

- 运行 sudo ./build/iot-qlinkapp.elf

```
1 dev[/dev/ttyUSB0]
2 [APP: main.c: 150] open cli function
3 open at uart succeed
4 [QLINK: cloud_qlink.c: 119] FW version: basic_AT_v2.1.2
5 [APP: main.c: 30] WiFi config in AP mode.....
6 [APP: main.c: 30] WiFi config in AP mode.....
7 [APP: main.c: 34] Wi-Fi connected
8 [APP: main.c: 42] Cloud connected
```

#### (4) 基于虚拟网关使用说明

- 按照4.3.1使用流程中说明搭建好本地虚拟网关开发环境
- 测试本地环境是否OK

如果本地环境OK，点击配网andlinkdemo1.0.apk 开始连接按钮，会有如下页面：



- 配网操作

通过cli串口发送 raw start命令 或者 按键触发调用qlink\_raw\_start()接口使设备进入本地网关一键配网模式，

同时手机APP点击开始连接按钮，绑定成功后APP提示添加成功，后台虚拟网关下面会显示出设备，如下：

## 已申请的设备

## 已入库的设备



**T1报警盒子**  
设备ID: CMCC-30531-...

所属产品: 30531-T1报警盒子

所属网关: C48E8F89B417

上级设备: 无

固件版本: 1.4.3

软件版本: 1.0.1

连接状态: 在线

注册时间: 2018-10-25 20:15:19

最近上线时间: 2018-10-25 20:15:20

- 数据下发

通过后台获取所有字段，操作如下所示：

### T1报警盒子/在线调试 ?

调试进度

历史数据

查看信息

返回列表

设备当前状态

连接状态: 在线    最近上报数据的时间: 2018-10-26 17:36:08    最近功能数据有改变的时间: 2018-10-26 17:36:08

### 设备控制

命令类型

请求上报所有功能值

发送命令

调用数据

```
{
```

通过后台获取部分字段，操作如下所示：

设备控制

❖ 命令类型

请求上报功能值

❖ 功能标识名

firmware

❖ 功能索引 ?

+

❖ 功能标识名

permitJoining

❖ 功能索引 ?

+

-

❖ 功能标识名

softVersion

❖ 功能索引 ?

+

-

发送命令

调用数据

```
{
  "parameters": {
    "param": [
      {
        "name": "firmware"
      },
      {
        "name": "permitJoining"
      },
      {
        "name": "softVersion"
      }
    ]
  }
}
```

通过后台下发命令，操作如下所示：

设备控制

❖ 命令类型

控制设备

❖ 功能标识名

permitJoining

❖ 功能值

1

❖ 功能索引 ?

+

发送命令

调用数据

```
{
  "parameters": {
    "param": [
      {
        "name": "permitJoining",
        "content": "1"
      }
    ]
  }
}
```

返回结果

```
{
  "resultCode": 0
}
```

设备收到云端获取字段指令后，需要调用mx\_status emh\_qlink\_send\_json\_to\_cloud( char \*type, uint8\_t \*data, uint32\_t len ) 函数将数据发送到云端

- 数据上传

mx\_status qlink\_send\_data(char \*type, uint8\_t \*data, uint32\_t len), 通过此函数 即可完成发送数据的功能

```
1 type: 为发送数据的类型, 注意主动上报和响应要选择不同类型
2
3         设备主动上报:           type = Inform
4         人为触发的数据上报:     type = ParamChange
5         云端请求上报:           type = Data
6         文件操作:               type = File
7 data: 上传具体内容
8
```

也可通过cli串口输入send type 发送数据

#### (5) 基于云网关使用说明

- 提交XML配网资料

完成本地网关验证后, 可以提交配网引导界面相关资料, 杭研配置XML到手机APP, 之后方可进行云网关验证操作,

提交配网资料可以提交本地网关+云网关方式资料或者仅通过云网关配网资料, 建议提供只用云网关配网资料,

如此一来, 上线验证只需要验证云网关部分即可, 减少验证流程, 提高上线时间

- 配网操作

通过cli串口发送 user start命令 或者 按键触发调用qlink\_user\_start()接口使设备进入云网关用户配网模式,

同时手机APP(resource/qlink/2.云网关/app路径)进入添加设备页面, 扫描到如下页面, 说明设备进入了配网页面:





点击弹出来的添加按钮即可进行配网操作

#### 4.3.3 注意事项

##### (1) 云端查询数据

回调函数`emh_ev_qlink_get_local_attrs()`收到数据时候，用户需要根据收到的内容选择发送给云端的数据，组成json包后再调用`qlink_send_data()`接口将数据发送给远端，注意类型应该是"ParamChange"类型。

##### (2) 云端下发命令数据

云端下发命令，回调函数`emh_ev_qlink_set_local_attrs()`会收到数据，用户需要根据字段的类型进行处理，如果字段即下发又要求上报，则return 1,否则return 0。

##### (3) 其他

联系人：庆科李博

邮箱： [libo@mxchip.com](mailto:libo@mxchip.com)

## 5. 移植SDK

- 只需要实现platform/mx\_hal.h文件中的函数即可