**统一SDK设计文档**

|  |  |  |  |
| --- | --- | --- | --- |
| **版本号** | **修订日期** | **修订内容** | **修订人** |
| V1.0 | 2016/7/26 | 新增 | 雷希;雷洪 |
| V1.1 | 2016/8/10 | 新增各协议模块必须实现接口定义 | lx |

1. 背景

为方便用户接入onenet，减少设备的接入成本，实现统一的接入流程，统一各协议接入功能接口，并多提供平台的移植支持，给用户侧提供简单易用的sdk.

1. 功能
2. 统一接口

* 统一引导获取服务器地址流程.
* 统一注册设备接口.
* 统一上传数据点接口.
* 统一处理平台下发指令接口.
* 统一指令回复接口.

1. 各协议差异化设计

各协议如有特有功能，需单独提供该协议特定接口.

1. 多平台支持

* 默认提供linux/windows两个平台的支持.
* 提供抽象平台移植接口,便于三方设备移植接入.

1. Sdk层次结构



1. Platform Abstracct Interface

操作系统相关接口定义以及操作系统的相应实现.

1. BASE

各协议公用的基础组件.

1. Protocol

不同协议的API以及相应实现.

1. Common

接入相关公用接口，协议公用接口.

1. 各层接口设计
2. Common层
   1. 注册设备接口.
   2. 获取服务地址接口.
   3. 鉴权接口.
   4. 上报数据点接口.
   5. 指令回复接口.
   6. 注册回调接口.

* 设备注册回调.
* 获取服务地址回调.
* 鉴权回调.
* 数据上报回调.
* 平台指令接收回调.
* 指令回复回调.

1. Protocol层
   1. EDP
   2. MQTT

* 连接鉴权接口
* 数据推送接口
* 订阅接口
* 取消订阅接口
  1. …

1. Base层
   1. 数据缓冲数据结构以及相关操作接口
   2. Channel传输抽象接口，底层socketapi由平台移植层提供.
2. 平台移植接口层
   1. socket相关操作接口.
   2. 获取系统时间接口.
   3. 系统sleep接口.
   4. 内存分配接口.
   5. 平台初始化接口.
   6. 平台清理接口.
3. SDK使用流程



1. 业务流程
2. 初始化SDK&建立连接



1. 数据发送流程.



1. 处理平台下发数据流程



1. 各协议必须实现的接口
2. int ont\_<protocol>\_create(ont\_device\_t \*\*dev)

创建协议实例。

参考实现如下：

struct ont\_<protocol>\_t {

ont\_device\_t dev;

/\* 协议相关的私有属性 \*/

}

如此定义的考虑是，减少动态分配，尽量使内存连续。

创建流程:

|  |
| --- |
| int ont\_<protocol>\_create(ont\_device\_t \*\*dev)  {  ont\_<protocol>\_t \*ins;  ins = (ont\_<protocol>\_t \*)ont\_platform\_malloc(sizeof(ont\_<protocol>\_t));  /\* 初始化协议相关的私有属性 \*/  \*dev = &ins->dev;  return ONT\_ERR\_OK;  } |

1. void ont\_<protocol>\_destroy(ont\_device\_t \*dev)

销毁实例。

1. int ont\_<protocol>\_connect(ont\_device\_t \*dev)

与服务器建立连接，包括完成鉴权等。

1. int ont\_<protocol>\_send\_datapoints(ont\_device\_t \*dev, const char \*payload, size\_t bytes)

发送数据点，payload已经按数据点格式8序列化。

1. ont\_device\_cmd\_t \*ont\_<protocol>\_get\_cmd(ont\_device\_t \*dev)

获取平台下发的命令，ont\_device\_cmd\_t 须由ont\_platform\_malloc创建。

1. int ont\_<protocol>\_reply\_cmd(ont\_device\_t \*dev, const char \*cmdid, const char \*resp, size\_t size)

发送命令响应数据。

1. int ont\_<protocol>\_keepalive(ont\_device\_t \*dev)

检查设备连接、发送心跳、接收平台下发数据等。