InHand VG710 车载网关设备开放平台

API 参考手册

资料版本：V1.1—2019.08

www.inhand.com.cn

北京映翰通网络技术股份有限公司

名词解释

缩写	名称	解释
DN	Device Networks	映翰通设备云平台
IDE	Integrated Development Environment	集成开发环境
SDK	Software Development Kit	软件开发包

RouterInfo模块

RouterInfo模块可用于获取路由器信息，如：拨号模块信息,GPS信息，路由器固件版本信息等。 该模块还可以用于通过CLI命令配置网关设备。

RouterInfo模块API使用示例代码如下：

```
# !/usr/bin/env python
# -*- coding: utf-8 -*-
'''
App example
@author: InHand
'''

import os
import time
import json
import logging
import libevent
from RouterInfo import RouterInfo
from AppTemplate_libevent import AppTemplate
from MessageChannel import MessagePush, INOS_PUSH_ADDR

INPY_APP_PATH = '/var/app/'

class AppExample(AppTemplate):
    def __init__(self, vendor_name, app_name, version='1.0.0'):
        AppTemplate.__init__(self, vendor_name, app_name, version,)
        self.ri = RouterInfo(self.base)
        self.timer = libevent.Timer(self.base, self.timer_handler,
userdata=None)
        self.app_name = app_name
        self.logger._logger.setLevel(logging.DEBUG)

    def timer_handler(self, evt, userdata):
        self.logger.info(self.ri.get_cellular_info()) # Get device cellular
information
        self.logger.info("*"*50)
        self.logger.info(self.ri.get_gps_info()) # GPS
        self.logger.info("*"*50)
        self.logger.info(RouterInfo.get_firmware_info())
```

```
        self.logger.info("*"*50)
        self.logger.info(RouterInfo.get_local_time())
        self.logger.info("time: %s" % time.time())
        self.timer.add(3)

    def init(self):
        self.running_conf = None
        self.timer.add(1)

if __name__ == '__main__':
    app = AppExample('InHand', '')
    app.init()
    app.run()
```

RouterInfo模块API详细描述如下：

class RouterInfo

实例化类对象

```
ri = RouterInfo(base)
```

参数

base: libevent.Base() 实例

类成员方法如下:

---

get_cellular_info()

参数

Null

接口描述

获取拨号模块信息

返回值

```
{
 "mcc": 460,
 "rssi": 26,
 "operator": "China Unicom",
 "iccid": "89860111811011851448",
 "mnc": 1,
 "imsi": "460011050002097"
 }
```

---

get_gps_info()

参数

Null

接口描述

获取GPS信息

返回值

```
{
    "gps_time": "2017-06-21T16:17:06+08:00",
    "latitude": 39.925755000000006,
    "speed": 0.407,
    "longitude": 116.19064833333334
}
```

> 其中speed单位为Knots(1knot = 1.852km/h)

---

get_dataflow()

参数

Null

接口描述

获取当前流量信息

返回值

```
{34237, 653456}
```

> 其中数组中的第一个 元素是接收的字节数，第二个元素是发送的字节数

---

get_local_time()

参数

Null

接口描述

获取设备当前时间

返回值

```
'2019-03-04T15:21:27+08:00'
```

---

get_firmware_info()

参数

Null

接口描述

获取路由器固件版本信息

返回值

```
{
  "lang":"Chinese",
  "hostname":"EdgeGateway",
  "timezone":"UTC-8",
  "model_name":"902B",
  "oem_name":"inhand",
  "bootloader":"2017.01.r10319",
  "serial_number":"IG9021900434193",
  "product_number":"EN00",
  "description":"www.inhandnetworks.com",
  "sw_version":"1.0.0.r10403",
  "encrypt_passwd":"no"
}
```


---

get_connect_device()

参数

Null

接口描述

获取与路由器连接的设备信息

返回值


```
[
    {"ip": "192.168.1.111", "mac": "b8:70:f4:da:cd:1b", "hostname": "", "dev":
"fastethernet 0/1"},
    {"ip": "192.168.30.30", "mac": "00:26:55:3a:68:b1", "hostname": "", "dev":
"bridge 1"}
 ]
```


---

get_dhcp_device()

参数

Null

接口描述

获取路由器DHCP管理的设备信息

返回值


```
 [
["br1","44:37:e6:59:8b:82","192.168.30.196","ES07734857","0 day, 23:21:53"]
 ]
```


---

get_app_info()

参数

Null

接口描述

获取设备安装App信息

返回值

```
[
    {"version": "0.0.1", "name": "RouterInfoSample", "sdkver": "1.3.2"}
]
```

---

set_openvpn_state(conf, state):

参数

conf 指OpenVPN的配置文件路径，需要填写绝对路径 state 设置OpenVPN的状态，若创建OpenVPN隧道则设置state='on'，否则设置state='off'

接口描述

创建/关闭OpenVPN隧道

返回值

Null

---

do_reboot():

参数

Null

接口描述

重启设备

返回值

Null

---

get_wlan_scan():

参数

Null

接口描述

获取wlan扫描信息

返回值

```
[
 {'signal': '-46','frequency': '2412','sec': '[WPA-PSK/AES][WPA2-
PSK/AES]','ssid': 'Inhand','bssid': '84:a9:c4:5f:6c:61'},
]
```

---

get_cert_status():

参数

Null

接口描述

获取当前证书申请的状态信息

返回值

```
{
    "status": "resume",
    "describe": "Resume Certificate",
    "timestamp": "2017-06-21T16:17:06+08:00"
}
```

---

get_fw_nat():

参数

Null

接口描述

获取当前firewall nat规则

返回值

```
['ip nat static 1.1.1.1 2.2.2.2', 'ip nat static 3.3.3.3 interface fastethernet 0/1']
```

---

get_connection_type():

参数

Null

接口描述

获取当前网络连接状态

返回值

1, 2, 3 or -1

> 1: cellular 2:ethernet 3:wifi -1:error

---

set_network_track(index, addr, frequency, timeout):

参数

index sla索引 addr 目的地址 frequency 探测频率 timeout 探测超时时间

接口描述

设置链路探测

返回值

True or False

> True: 成功 False: 失败

---

get_network_track():

参数

Null

接口描述

获取链路探测状态

返回值

```
[{'tid': 1, 'stat': 0}]
```

> tid : track id; stat: {0: 未活动; 1:活动; -1:出错}

InDB模块

InDB为用户提供了数据缓存操作的类封装API，示例代码如下：

```
import InDB

if __name__ == '__main__':
    dbPath = '/var/app/data/test.db'
    db = InDB.InDB(dbType=InDB.DB_TYP_SHARED,dbRole=InDB.DB_ROL_OWNER)
    db.open(dbPath)
    # 初始化表
    data_tb = db.get_table('data', tblType=InDB.TBL_TYP_KV)
  # key = "1" # py2
  key = b"1" # py3
    # 存入数据
    data_tb.put(key, "hello_ChengDu")
    # 取出数据
    print data_tb.get(key)
```

详细的描述如下:

class InDB

初始化参数

```
dbType:
    - DB_TYP_PRIVATE:
    this DB is only accessed by the own process itself. Note that any persistent
DB will be accessed by the system manage process. See the dbPersistent arg
below.
    - DB_TYP_SHARED:
    this DB is used to shard data with multiplie processes.

dbRole:
    - DB_ROL_OWNER:
    the present process own this database. It's the owner's responsibility to
create database and tables before the others using them. For a private DB, the
one who open it is forced to be owner.
    - DB_ROL_WRITER:
    the one who can write or read this database.
    - DB_ROL_READER:
    the one who can only read this database.
```

操作示例

```
db = InDB.InDB(dbType=InDB.DB_TYP_SHARED, dbRole=InDB.DB_ROL_OWNER)
```

打开数据库

open(self, dbPath)

参数

dbPath: 数据库存储路径

返回值

Null

操作示例

```
db.open(dbPath)
```

判断数据库是否打开

is_opened(self)

参数

Null

返回值

True or False

> True: 打开 False: 关闭

操作示例

```
db.is_opened()
```

关闭数据库

close(self)

参数

Null

返回值

Null

操作示例

```
db.close()
```

获取数据库表

get_table(self, name, tblType)

参数

```
name: 数据表名

tblType:
    - TBL_TYP_KV: 键值对类型数据表
    - TBL_TYP_TS: 时间序列类型表
```

返回值

表实例

操作示例

```
data_tb = db.get_table('data', tblType=InDB.TBL_TYP_KV)
```

插入数据

put(self, key, val)

参数

key：需要存储数据的关键字 Value：需要存储的数据, json格式

返回值

None

操作示例

```
key = b'power'
val={'addr':40001, 'timestamp':12341234, 'value':321}
data_tb.put(key, val)
```

读取数据

get(self, key)

参数

key: 获取存储数据的关键字

返回值

关键字对应的存储数据,无数据返回None

操作示例

```
key = b'power'
val = db.get(key)
```

出栈数据

pop(self, key)

参数

key：需要存储数据的关键字

返回值

关键字对应的存储数据,无数据返回None

操作示例

```
key = b'power'
val = db.pop(key)
```

删除数据

delete(self, key)

参数

key：需要删除数据的关键字

返回值

Null

操作示例

```
key = b'power'
db.delete(key)
```

获取数据库中所有key

keys(self)

参数

Null

返回值

存储数据key值列表

操作示例

```
db.keys()
```