

# IOVBox 服务接口开发文档

易通星云 内部资料

易通星云（北京）科技发展有限公司

2018.7

## 修订纪录

修改日期	修改内容	描述	修改人
2018/3/9	新增接口		周小波
2018/4/11	修改接口从 dbus 变成 js		陈莞
2018/7/5	新增接口描述		陈莞

## 目录

IOVBox 服务接口开发文档 .....	1
修订纪录.....	2
目录.....	3
1 引言.....	5
1.1 编写目的.....	5
1.2 名词定义.....	5
2 JS 接口说明.....	7
2.1 dbus_app 模块.....	7
2.1.1 接口定义.....	7
2.1.2 使用方式.....	8
2.2 GPS 模块.....	8
2.2.1 接口定义.....	8
2.2.2 使用方式.....	9
2.3 GPIO 模块.....	10
2.3.1 接口定义.....	10
2.3.2 使用方式.....	11
2.4 IMU 模块.....	11
2.4.1 接口定义.....	11

2.4.2	使用方式 .....	12
2.5	DIAL 模块 .....	12
2.5.1	接口定义 .....	12
2.5.2	使用方式 .....	14
2.6	LED 模块 .....	14
2.6.1	接口定义 .....	14
2.6.2	使用方式 .....	15
2.7	POWER 模块 .....	15
2.7.1	接口定义 .....	15
2.7.2	使用方式 .....	16
2.8	WIFI 模块 .....	16
2.8.1	接口定义 .....	16
2.8.2	使用方式 .....	18
2.9	BLUETOOTH 模块 .....	18
2.9.1	接口定义 .....	18
2.9.2	使用方式 .....	19
2.10	ADC 模块 .....	19
2.10.1	接口定义 .....	19
2.10.2	使用方式 .....	20

重要申明.....	21
注意.....	21
无担保声明.....	22

## 1 引言

### 1.1 编写目的

本文档的目的是阐述 iovBOX 中所提供 JS 功能服务接口的说明。

### 1.2 名词定义

序号	英文名称	备注
1	JS	JavaScript 简称

2	D-Bus	数据总线
3		
4		
5		

易通星云 版权所有

## 2 JS 接口说明

软件总体采用服务模型，各个 JS 接口通过服务 D-Bus 总线为应用提供服务和数据通道。

### 2.1 dbus\_app 模块

dbus\_app 模块是提供 dbus 总线服务的模块。JavaScript 应用可以调用 dbus\_app 模块提供的接口进行 dbus 总线服务的初始化，其他服务在 dbus 总线上的申请注册。

#### 2.1.1 接口定义

**register\_app\_name()**

函数意义：初始化所有服务。当其他服务已经在 dbus 总线服务上注册以后，使用此接口来进行初始化。

返回值：null

**listener\_fun\_array**

属性意义：其他服务在 dbus 总线服务上注册的列表

返回值：null

**proc**

属性意义：dbus 总线服务的 promise 实体，保证所有服务注册并初始化完成后，才提供服务。

返回值：null

## 2.1.2 使用方式

```
var dbus_app = require('dbus-js/dbus_app.js');  
var dbus_obj = new dbus_app();
```

生成 dbus 服务总线。

```
var gpslib = require('dbus-js/gps.js');  
var gpiolib = require('dbus-js/gpio.js');  
var gpio = new gpiolib(dbus_obj);  
var gps = new gpslib(dbus_obj);
```

在 dbus 服务总线上申请注册 gps 服务, 申请注册 gpio 服务。

```
dbus_obj.register_app_name();
```

dbus 服务初始化, 会将已经申请的服务统一初始化。

## 2.2 GPS 模块

GPS 模块是提供 GPS 定位点的采集服务。JavaScript 应用可以调用 GPS 模块提供的接口进行查询 GPS 定位点和控制 GPS 模块功能, 也可以监听 GPS 定位点上报事件。

### 2.2.1 接口定义

```
onReportNMEA(inputCallback)
```

函数意义: 当 GPSD 上报定位数据的时候, 触发 inputCallback 处理数据

返回值: NMEA string

```
onReportData(inputCallback)
```

函数意义: 当 GPSD 上报定位数据的时候, 触发 inputCallback 处理数据

返回值: {code, message, result}

```
getinfo(outputCallback)
```

函数意义: 从 GPSD 查询定位数据, 查询完成触发 inputCallback 处理数据



返回值: {code, message, result}

#### `setReportType(type, outputCallBack)`

函数意义: 设置 GPSD 上报数据的格式 type

参数 1: 上报数据格式 1=NMEA 2=JSON

返回值: {code, message, type, result}

#### `setReportEnable(enable, outputCallBack)`

函数意义: 设置 GPSD 是否主动上报数据

参数 1: 1 = 主动上报 0 = 被动查询

返回值: {code, message, enable, result}

#### `setDebugLevel(level, outputCallBack)`

函数意义: 设置 GPSD 日志等级

参数 1: GPSD 日志等级

返回值: {code, message, level, result}

其中

code: 0=success -1=fail

message: dbus 返回值

result: dbus\_err or JSON result

### 2.2.2 使用方式

```
var gpslib = require('dbus-js/gps.js');  
var gps = new gpslib(dbus_obj);
```

在 dbus 服务总线上申请 gps 服务, 当需要申请其他服务的时候, 可以一并申请。

```
dbus_obj.register_app_name();
```

dbus 服务初始化, 会将已经申请的服务统一初始化。

```
gps.onReportNMEA(function(res) {  
  console.log('report nmea :'+ (res));  
});
```

调用 GPS 接口。

## 2.3 GPIO 模块

GPIO 模块是提供 IO 控制和采集服务。其中 GPIO 共包括 5 个输入通道和 2 个输出通道。JavaScript 应用可以调用 GPIO 模块提供的接口进行查询和控制，也可以监听事件。

### ■ 输入通道：

- 1) "Input IO1"
- 2) "Input IO2"
- 3) "Input IO3"
- 4) "Input IO4"
- 5) "Input ACC"

### ■ 输出通道：

- 1) "chan0"
- 2) "chan1"

### 2.3.1 接口定义

```
onInputChange(inputCallback(res))
```

函数意义：当输入通道内的 IO 变化时，上报所有输入通道的 IO 值。

输出：{IO1,IO2,IO3,IO4,ACC}.

```
setLow(channel, outputCallback(res))
```

函数意义：设置输出通道 channel 的 IO 为低，并在成功后回调。

输出：{code, message, channel, result}

```
setHigh(channel, outputCallback(res))
```

函数意义：设置输出通道 channel 的 IO 为高，并在成功后回调。

输出：{code, message, channel, result}

```
getCurrentIOState(inputCallback(res))
```

函数意义：获取当前情况下所有输入通道 IO 值。

输出：{code, message, result}

```
setDebugLevel(level, outputCallback(res))
```

函数意义：设置 GPIO 模块的调试日志类别。

参数：调试日志类别

返回值：{code, message, level, result}

### 2.3.2 使用方式

与 GPS 模块流程一致，接口略微不同。（略）

## 2.4 IMU 模块

IMU 模块是提供配置陀螺仪，以及读取陀螺仪数据输出的模块。JavaScript 应用程序可以调用 IMU 模块接口完成开启/关闭陀螺仪上报功能、改变调试等级、监听陀螺仪上报数据、查询陀螺仪测量数据的功能。

### 2.4.1 接口定义

```
onIMUReport(outputCallback(res))
```

函数意义：每当 IMU 有数据上报的时候，调用回调处理上报数据

返回值：{Gyro\_x, Gyro\_y, Gyro\_z, Acc\_x, Acc\_y, Acc\_z}.

```
setSample(sec, outputCallback(res))
```

函数意义：设置 IMU 的上报间隔。

参数：设置多少 ms 上报一次。

返回值：{code, message, sec, result}

```
setSwitch(enable, outputCallback(res))
```

函数意义：打开 IMU 的上报功能

参数： 1 代表打开，0 代表关闭

返回值：{code, message, enable, result}

```
setDebugLevel(level, outputCallback(res))
```

函数意义：设置 IMU 的调试日志类别。

参数：调试日志类别

返回值：{code, message, level, result}

## 2.4.2 使用方式

与 GPS 模块流程一致，接口略微不同。(略)

## 2.5 DIAL 模块

DIAL 模块是提供配置查询 4G 模块，以及通过 4G 模块读取发送数据的功能。

JavaScript 应用程序可以调用 DIAL 模块接口完成发送消息，读取消息列表，删除消息、改变调试等级，查询信号，查询模块状态等功能。

### 2.5.1 接口定义

```
onStateChange(inputCallback(res))
```

函数意义：每当 4G 模块状态变化时，调用回调处理上报数据

返回值：Dial\_State

- 1:"dial init",
- 2:"dial start",
- 3:"dial running",
- 4:"dial stopping",
- 5:"dial stopped",

6:"dial none"

`onSignalChange(inputCallback(res))`

函数意义：每当 4G 模块信号强度发生变化时，调用回调处理上报数据

返回值：string.

`onSimStateChange(inputCallback(res))`

函数意义：每当 4G 模块的 sim 卡状态发生变化，调用回调处理上报数据

返回值：TRUE/FALSE.

`onNetworkChange(inputCallback(res))`

函数意义：每当 4G 模块的 sim 卡状态发生变化，调用回调处理上报数据

返回值：{dail\_info}

`sendMessage(phone, msg, outputCallback(res))`

函数意义：从 4G 模块发送消息到指定电话号码。

参数 1：电话号码，字符串。

参数 2：消息，字符串。

返回值：{code, message, phone, msg, result}

`getInfo(outputCallback(res))`

函数意义：获取 4G 模块的整体状态

返回值：{code, message, result/dial\_info}

`getMessageList(status,outputCallback(res))`

函数意义：获取 4G 模块的消息列表

返回值：{code, message, status, result}

`deleteMessage(msgid, outputCallback(res))`

函数意义：删除指定消息编号的消息。

返回值：{code, message, msgid, result}

`setDebugLevel(level, outputCallback(res))`

函数意义：设置 DIAL 模块的调试日志类别。

参数：调试日志类别

返回值：{code, message, level, result}

```
getHWInfo(outputCallback(res))
```

函数意义：获取 4g 模块的硬件信息。

参数：调试日志类别

返回值：{code, message, result=hwinfo}

## 2.5.2 使用方式

与 GPS 模块流程一致，接口略微不同。（略）

## 2.6 LED 模块

LED 模块是提供控制 IOVBox 上 LED 的模块。LED 包括 4 个，颜色包括 black, blue, cyan, orange, red, white, pink, green。

### 2.6.1 接口定义

```
setColor(ledid, color, inputCallback(res))
```

函数意义：设置对应标识号的 LED，显示特定的颜色

参数 1：led 的标识号 0~4

参数 2：颜色 black,blue,cyan,orange,red,white,pink,green

返回值：{code, message, id, color, result}

```
setDebugLevel(level, outputCallback(res))
```

函数意义：设置 LED 模块的调试日志类别。

参数：调试日志类别

返回值: {code, message, level, result}

## 2.6.2 使用方式

与 GPS 模块流程一致, 接口略微不同。(略)

## 2.7 POWER 模块

power 模块是电源管理相关的安全策略配置模块, JavaScript 应用程序可以通过 Power 模块的接口, 完成配置调试等级, 查询主备电压, 上报主备电压告警, 设置休眠周期, 设置工作模式等等功能。

### 2.7.1 接口定义

```
setData(type, data, inputCallback(res))
```

函数意义: 向 POWER 模块设置电源管理参数。

参数 1: 电源管理参数类型

参数 2: 电源管理参数的值

返回值: {code, message, type, data, result}

```
getData(type, inputCallback(res))
```

函数意义: 获取 POWER 模块的电源管理参数。

参数 1: 电源管理参数类型

返回值: {code, message, type, result, data}

```
onWarnInfo(inputCallback(res))
```

函数意义: 获取 POWER 模块中特定电源管理参数的告警信息。

参数 1: 电源管理参数类型

返回值: warninfo

```
0:'MAIN_POWER_SLEEP',  
1:'MAIN_POWER_VLOW',  
2:'MAIN_POWER_OFF',  
3:'BACK_POWER_VLOW',  
4:'DEVICE_UNCOVER',  
5:'DEVICE_ACC_ON',  
32:'MAIN_POWER_NORMAL',  
33:'MAIN_POWER_VHIGH',  
34:'MAIN_POWER_ON',  
35:'BACK_POWER_VHIGH',  
36:'DEVICE_COVER',  
37:'DEVICE_ACC_OFF',  
64:'SD_CARD_REMOVE',  
65:'MOBILE_NET_WAKEUP'
```

```
setDebugLevel(level, outputCallback(res))
```

函数意义：设置 POWER 模块的调试日志类别。

参数：调试日志类别

返回值：{code, message, level, result}

## 2.7.2 使用方式

与 GPS 模块流程一致，接口略微不同。（略）

## 2.8 WIFI 模块

wifi 模块是 wifi 配置模块，JavaScript 应用程序可以通过 wifi 模块的接口，完成配置调试等级，设置 AP，搜索 AP，设置 STA 等等功能。

### 2.8.1 接口定义

```
onWifiStatus(outputCallback)
```

函数意义：连接状态报告。

返回值：{status}



0: "DISCONNECTED",  
1: "CONNECTED"

```
queryWifiInfo(outputCallback)
```

函数意义：查询 WIFI 报告

返回值：{code, message, result/wifi\_info}

```
setModeAP(opts, inputCallback(res))
```

函数意义：WIFI 设置 AP

参数 1: opts{{string} ssid; {string} password; {bool} hidden; }

返回值：{code, message, opts, result}

```
setModeSTA(opts, outputCallback)
```

函数意义：WIFI 设置 STA

参数：opts {{string} ap\_name; {string} password}

返回值：{code, message, opts, result}

```
setModeAPSTA(opts, outputCallback)
```

函数意义：WIFI 设置 APSTA

参数：opts{{string} ssid; {string} password; {bool} hidden; {string} ap\_name;  
{string} password}

返回值：{code, message, opts, result}

```
ScanWiFiAccessPoint(outputCallback)
```

函数意义：搜索 WIFI 的 AP

返回值：{code, message, result}

```
setDebugLevel(level, outputCallback(res))
```

函数意义：设置 WIFI 模块的调试日志类别。

参数：调试日志类别

返回值：{code, message, level, result}

## 2.8.2 使用方式

与 GPS 模块流程一致，接口略微不同。（略）

## 2.9 BLUETOOTH 模块

BLUETOOTH 模块是蓝牙管理配置模块，JavaScript 应用程序可以通过 BLUETOOTH 模块的接口，完成配置调试等级，查询版本，发送消息等等功能。

### 2.9.1 接口定义

`onConnectStatus(outputCallback)`

函数意义：连接状态报告

返回值：{connectstatus}

- 1:"CONNECT\_STATUS\_IDLE",
- 2:"CONNECT\_STATUS\_FAST",
- 3:"CONNECT\_STATUS\_SLOW",
- 4:"CONNECT\_STATUS\_CONNECTED",
- 5:"CONNECT\_STATUS\_CONNECTED\_AND\_BONDED",
- 6:"CONNECT\_STATUS\_UNKNOWN"

`onWorkStatus(outputCallback)`

函数意义：工作状态报告

返回值：{workstatus}

- 1:"WORK\_STATUS\_NONE",
- 2:"WORK\_STATUS\_RUNNING",
- 3:"WORK\_STATUS\_STOP"

`onMessage(outputCallback(res))`

函数意义：消息数据报告

返回值：{messagebody}

`setServerEnable(enable, outputCallback)`

函数意义：开启/关闭功能

参数: opts {{string} ap\_name; {string} password}

返回值: {code, message, enable, result}

`sendMessage(message, outputCallback)`

函数意义: 发送消息数据

参数: message

返回值: {code, message, (content), result}

`setDebugLevel(level, outputCallback)`

函数意义: 设置 BLUETOOTH 模块的调试日志类别。

参数: 调试日志类别

返回值: {code, message, level, result}

## 2.9.2 使用方式

与 GPS 模块流程一致, 接口略微不同。(略)

## 2.10 ADC 模块

ADC 模块是 ADC 管理配置模块, JavaScript 应用程序可以通过 ADC 模块的接口, 完成配置调试等级, 上报 ADC, 设置采样周期, 开关采样等等功能。

### 2.10.1 接口定义

`onADCReport(outputCallback)`

函数意义: ADC 上报接口

返回值: {AD0, AD1}

`setSample(outputCallback)`

函数意义: 设置采样周期

参数 1: `sec >= 50ms`

返回值: `{code, message, sec, result}`

```
setSwitch(enable, outputCallback(res))
```

函数意义: 消息数据报告

参数 1: `enable`

返回值: `{code, message, enable, result}`

```
setDebugLevel(level, outputCallback)
```

函数意义: 设置 ADC 模块的调试日志类别。

参数: 调试日志类别

返回值: `{code, message, level, result}`

## 2.10.2 使用方式

与 GPS 模块流程一致, 接口略微不同。(略)

## 重要申明

易通星云（北京）科技发展有限公司为客户提供全方位的技术支持，用户可与就近的易通办事处、分公司、代理商联系，也可直接与公司总部联系。

### 易通星云（北京）科技发展有限公司

北京市朝阳区霄云路3号中科霄云大厦309室

公司总机：010-64612620

网 址：[www.beidouapp.com](http://www.beidouapp.com)

客户服务邮箱：[support@beidouapp.com](mailto:support@beidouapp.com)

易通公司保留对产品外观及设计改变的权利，恕不另行通知。

**版权所有©易通星云（北京）科技有限责任公司 2015。保留一切权利。**

非经易通星云（北京）科技发展有限公司书面同意，任何单位和个人不得擅自摘抄、复制本手册内容的部分或全部，并不得以任何形式传播。

本手册中描述的产品，可能包含易通星云（北京）科技发展有限公司及其可能存在的许可人享有版权的软件，除非获得相关权利人的许可，否则，任何人不能以任何形式对前述软件进行复制、分发、修改、摘录、反编译、反汇编、解密、反向工程、出租、转让、分许可以及其他侵犯软件版权的行为。

### 注意

本手册描述的产品及其附件的某些特性和功能，取决于当地网络的设计和性能，以及您安装的软件。某些特性和功能可能由于当地网络运营商或网络服务供应商不支持，或者由于当地网络的设置，或者您安装的软件不支持而无法实现。因此，

本手册中的描述可能与您购买的产品或其附件并非完全一一对应。

易通星云（北京）科技发展有限公司保留随时修改本手册中任何信息的权利，无需进行任何提前通知且不承担任何责任。

## 无担保声明

本手册中的内容均“如是”提供，除非适用法律要求，易通星云（北京）科技发展有限公司对本手册中的所有内容不提供任何明示或暗示的保证，包括但不限于适销性或者适用于某一特定目的的保证。

在法律允许的范围内，易通星云（北京）科技发展有限公司在任何情况下，都不对因使用本手册相关内容而产生的任何特殊的、附带的、间接的、继发性的损害进行赔偿，也不对任何利润、数据、商誉或预期节约的损失进行赔偿。