

# LuatBoard 开发板通信协议 V1.3

## 一、约定

开发板通讯协议：MQTT 3.1.x 文件编码格式：UTF-8 , JSON

- 小程序发布主题：/luatask/demo/cmd/868575028802041
- 小程序订阅主题：/luatask/demo/rep/868575028802041
- 主板和小程序交互报文定义：

指令(cmd)	指令类型(type)	指令参数(ext)	含义
Demo	ui	无或自定义	UIdemo 演示
	update	nil	版本升级演示,reboot 参数表示升级后重启
		reboot	
	i2c	无	AM2320 温湿度传感器演示
	qrcode	无或自定义	二维码自动生成演示
	aduiio	1 或 2 或 3	播放 MP3 演示
	led	1 或 2 或 3	LED 灯控制演示
	lbs	无	基站定位 UI 演示
	ntp	无	NTP 演示支持 UI 显示
	io	0	远程控制 IO33 输出高电平或者低电平
		1	
Read	status	无	读取 LuatBoard 的状态信息
	serverconf	无	读取 LuatBoard 的默认配置信息
Syscmd	reboot	无	服务器远程重启开发板
	upfreq	1-1440	开发板默认上报频率，单位分钟
	fly	0	服务器远程控制开发板进飞行模式
		1	开发板进入飞行模式 1 分钟后自动退出
shell	sendat	AT 指令字符串	远程下发 AT 指令
	print	代码字符串	演示在线编程

注：所有下发都会返回状态信息,指令字符串全部为小写字母

## 二、下行指令范例

### ui:

小程序提供选择框：

#### 1、默认：

JSON 范例：

```
{  
    "cmd":"demo",  
    "type":"ui",  
    "ext":"logo"  
}
```

#### 2、自定义输入：

str1	8 个中文字符或者 16 个英文字符
str2	8 个中文字符或者 16 个英文字符
str3	8 个中文字符或者 16 个英文字符
str4	8 个中文字符或者 16 个英文字符

JSON 范例：

```
{  
    "cmd":"demo",  
    "type":"ui",  
    "ext":{  
        "str1":"str1...",  
        "str2":"str2...",  
        "str3":"str3...",  
        "str4":"str4..."  
    }  
}
```

## led:

小程序提供选择框:

流水灯 1

心跳灯 2

等级灯 3

组合灯 4

JSON 范例(流水灯):

```
{  
  "cmd": "demo",  
  "type": "led",  
  "ext": 1  
}
```

## qrcode:

默认显示"<http://www.openluat.com>" 的二维码,如果你需要自定义二维码的内容请输入:

JSON 范例:

```
{  
  "cmd": "demo",  
  "type": "qrcode",  
  "ext": "http://wiki.openluat.com"  
}
```

## i2c:

I2C 使用外设 AM2320 演示,请按提示,将 AM2320 插入 I2C 扩展口,如果你插上了 SPI

OLED 屏,会在屏幕上同步显示 AM2320 的结果

开发板返回: AM2320 的温湿度信息

JSON 范例:

```
{
```

```
"cmd":"demo",  
"type":"i2c",  
"ext":""  
}
```

## update:

演示远程更新 script 功能

小程序先下发查询指令，查询当前状态 status，包括信号强度，版本信息，和 IMEI 。

然后下发远程升级指令，

小程序提供选择框：更新后是否重启模块(nil 或 “reboot”)

然后 让用户选择是否重启模块的指令，

重启模块后会自动上报最新的状态信息，对比框提示新旧版本号差异

JSON 范例：

```
{  
  "cmd":"demo",  
  "type":"update",  
  "ext":"reboot"  
}
```

## lbs:

演示基站定位能力

小程序下发指令后，开发板进行 LBS 定位，并返回坐标。小程序看情况是否调用地图显示？？？

下发指令后，屏幕上会显示基站定位的坐标和地址

JSON 范例：

```
{  
  "cmd":"demo",  
  "type":"lbs",  
  "ext":""  
}
```

```
}
```

## ntp:

演示上电时间和 NTP 执行后的时间

指令 `cmd = "demo",type="ntp",`

屏幕上会显示当前时间，3 秒后小程序主动查询状态，可以获得同步后的时间

JSON 范例：

```
{  
  "cmd": "demo",  
  "type": "ntp",  
  "ext": ""  
}
```

## audio:

开发板自带三首 mp3 音乐 `pwron.mp3,sms.mp3,call.mp3`，小程序应该提供三个选择框让用户选择播放的 MP3，然后下发指令。

JSON 范例：

```
{  
  "cmd": "demo",  
  "type": "audio",  
  "ext": "pwron"  
}
```

## io:

小程序演示远程控制设备开关，这里的设备用 IO33 脚对应的红色 LED 灯代替，小程序上建议用个 LED 图标点亮 1 和灭 0 的状态与下发指令同步显示。

JSON 范例：

```
{
```

```
"cmd": "demo",  
"type": "io",  
"ext": 1  
}
```

## 指令为 read 的情况

小程序提供按钮，由用户点击，下发查询，并回显开发板返回的结果（主意数据格式化显示）。

查询状态 JSON 范例：

```
{  
  "cmd": "read",  
  "type": "status",  
  "ext": ""  
}
```

查询服务器配置信息 JSON 范例：

```
{  
  "cmd": "read",  
  "type": "serverconf",  
  "ext": ""  
}
```

## 上报状态：

JSON 范例

```
{  
  "cmd": "demo",  
  "type": "qrcode",  
  "csq": 11,  
  "softVer": "0.0.1",  
}
```

```
"lodVer": "Luat_V0025_8955_UI",  
"lng": "114.6044148"  
"lat": "033.8566738",  
"addr": "河南省郑州市中原区金河路",  
"stamp": "20180701125922",  
"ext": {  
    temp = "25.0"  
    hum = "45.9%"  
},  
}
```

**主意：ext** 只有在需要上报返回数据的时候才会有数据

## 上报服务器信息：

JSON 范例：

```
{  
    "sub": "/luatask/demo/cmd/",  
    "pub": "/luatask/demo/rep/",  
    "ip": "47.96.86.11",  
    "port": 1883,  
    "lastwill": "/luatask/demo/offline/",  
    "pwd": "",  
    "uid": "",  
    "qos": 0,  
    "keepalive": 60  
}
```

# 指令为 syscmd 的情况

小程序提供按钮下发，由用户自己监测开发板情况

## 远程重启开发板：

JSON 范例：

```
{  
  "cmd": "syscmd",  
  "type": "reboot",  
  "ext": ""  
}
```

## 设置 LuatBoard-Air202 自动上报频率

小程序下发默认上报频率，默认上报频率是 1 分钟 1 次。

JSON 范例：

```
{  
  "cmd": "syscmd",  
  "type": "upfreq",  
  "ext": "5"  
}
```

## 远程控制开发板飞行模式：

远程下发飞行模式指令，当 ext 为 1 的时候，1 分钟后会自动退出飞行模式。

JSON 范例：

```
{  
  "cmd": "syscmd",  
  "type": "fly",  
  "ext": 1  
}
```



# 指令为 shell 的情况

分 sendat 和 print 的情况

## sendat:

JSON 范例:

```
{  
  "cmd": "shell",  
  "type": "sendat",  
  "ext": "用户的 AT 指令"  
}
```

## printf:

JSON 范例:

```
{  
  "cmd": "shell",  
  "type": "printf",  
  "ext": "用户的 lua 代码"  
}
```