

# spec串口指令调用介绍

## 概述

MIIO模组负责串口指令到JSON RPC的转换，本文档仅供了解RPC定义，MCU端可无需关心具体实现。

RPC中method "get\_properties"、"set\_properties"和"properties\_changed"均为对属性操作，对应设备基于miot-spec的功能定义中属性的access。而"action"和"event\_occured"对应设备实例中方法和事件的定义。

RPC method	设备miot-spec功能定义
get_properties	可读属性，access:[read]
set_properties	可写属性，access:[write]
properties_changed	变化上报属性，access:[notify]
action	方法定义（action）
event_occured	事件定义(event)

## spec串口指令定义

### 读属性：get\_properties

#### 命令格式

```
get_properties <siid> <piid> ... <siid> <piid>
```

#### 应答格式

```
<siid> <piid> <code> [value] ... <siid> <piid> <code> [value]
```

#### 示例

#### iid-json-rpc 映射 串口rpc

```
iid-json-rpc请求包：

{
  "id" : 123,
  "method" : "get_properties",
  "params" : [
```

```

        params : [
            {
                "did" : "1234" ,
                "siid" : 1,
                "piid" : 2
            },
            {
                "did" : "1234" ,
                "siid" : 1,
                "piid" : 3
            }
        ]
    }
}

```

串口rpc请求包:

```

                                | siid | piid | siid | piid |
get_properties      1         2         1         3

```

iid-json-rpc应答包（成功）:

```

{
  "id" : 123,
  "result": [
    {
      "did" : "1234",
      "siid" : 1
      "piid" : 2,
      "value": 10,
      "code" : 0
    },
    {
      "did" : "1234",
      "siid" : 1
      "piid" : 3,
      "value": "hello",
      "code" : 0
    }
  ]
}

```

串口rpc应答包（成功）

```

    | siid | piid | code | value | siid | piid | code | value |
    1     2     0     10     1     3     0     "hello"

```

iid-json-rpc应答包（有失败）:

```

{
  "id" : 123,
  "result": [
    {
      "did" : "1234",

```

```
        "siid" : 1
        "piid" : 2,
        "value": 10,
        "code" : 0
    },
    {
        "did"  : "1234",

        "siid" : 1
        "piid" : 3,
        "code" : -4001
    }
]
}
```

串口rpc应答包（失败）

	siid		piid		code		value		siid		piid		code	
	1		2		0		10		1		3		-4001	

串口命令执行get\_properties

步骤	串口命令
1	↑ get_down
2	↓ down get_properties 1 2 1 3
3	↑ result 1 2 0 10 1 3 0 "hello"
或	
3	↑ result 1 2 0 10 1 3 -4001

写属性：set\_properties

命令格式

set\_properties <siid> <piid> <value> ... <siid> <piid> <value>

应答格式

<siid> <piid> <code> ... <siid> <piid> <code>

示例

iid-json-rpc 映射 串口rpc

-----

iid-json-rpc请求包:

```
{
  "id": 123,
  "method" : "set_properties",
  "params" : [
    {
      "did" : "1234",
      "siid" : 1,
      "piid" : 1,
      "value": 10
    },
    {
      "did" : "1234",
      "siid" : 1,
      "piid" : 88,
      "value": "hello"
    }
  ]
}
```

串口rpc请求包:

	siid	piid	value	siid	piid	value
set_properties	1	1	10	1	88	"hello"

-----

iid-json-rpc应答包:

```
{
  "id": 123,
  "result": [
    {
      "did" : "1234",
      "siid" : 1,
      "piid" : 1,
      "code" : 0          // 成功
    },
    {
      "did" : 1234,
      "siid" : 1,
      "piid" : 88,
      "code" : -4003      //属性不存在
    }
  ]
}
```

串口rpc应答包

	siid	piid	code	siid	piid	code
--	------	------	------	------	------	------

110188-4003

-----

串口命令执行set\_properties

步骤	串口命令
1	↑ get_down
2	↓ down set_properties 1 1 10 1 88 "hello"
3	↑ result 1 1 0 1 88 -4003

执行动作：action

命令格式

action <siid> <aiid> <piid> <value> ... <piid> <value>

应答格式

<siid> <aiid> <code> <piid> <value> ... <piid> <value>

示例

iid-json-rpc 映射 串口rpc

-----

iid-json-rpc请求包:

```
{
  "id": 123,
  "method" : "action",
  "params" :
  {
    "did" : "1234",
    "siid" : 1,
    "aiid" : 1,
    "in" : [
      {
        "piid" : 1,
        "value": 10
      }
    ]
  }
}
```

串口rpc请求包:

	siid	aiid	piid	value
action	1	1	1	10

iid-json-rpc应答包（成功）:

```
{
  "id" : 123,
  "result" : {
    "code" : 0,
    "out" : [
      {
        "piid" : 3,
        "value" : 10
      }
    ]
  }
}
```

串口rpc应答包（成功）

	siid	aiid	code	piid	value
	1	1	0	3	10

iid-json-rpc应答包（失败）:

```
{
  "id" : 123,
  "result" : {
    "code" : -4004,
  }
}
```

串口rpc应答包（失败）

	siid	aiid	code
	1	1	-4004

串口命令执行action

步骤	串口命令
1	↑ get_down
2	↓ down action 1 1 1 10
3	↑ result 1 1 0 3 10
或	
3	↑ result 1 1 -4004

属性上报：properties\_changed

命令格式

```
properties_changed <siid> <piid> <value> ... <siid> <piid> <value>
```

应答格式

```
无（应答对设备意义不大）
```

示例

iid-json-rpc 映射 串口rpc

```
-----

iid-json-rpc请求包:

{
  "id": 123,
  "method": "properties_changed",
  "params": [
    {
      "did" : "1234",
      "siid" : 1,
      "piid": 1,
      "value": 17
    },
    {
      "did" : "1234",
      "siid" : 1,
      "piid": 1,
      "value": "hi"
    }
  ]
}
```

串口rpc请求包：

```
      | siid | piid | value | siid | piid | value |
properties_changed    1      1      17      1      1      "hi"
```

-----

串口命令执行properties\_changed

步骤	串口命令
1	↑ properties_changed 1 1 17 1 1 "hi"
2	↓ ok
或	
2	↓ error

事件上报：event\_occured

命令格式

```
event_occured <siid> <eiid> <piid> <value> ... <piid> <value>
```

应答格式

无（应答对设备意义不大）

示例

iid-json-rpc 映射 串口rpc

-----

iid-json-rpc请求包：

```
{
  "id": 123,
  "method": "event_occured",
  "params":
  {
    "did": "1234",
    "siid": 1,
    "eiid": 1,
    "arguments":
```



```

    [ //事件参数表
      {
        "piid": 1,
        "value": 17
      },
      {
        "piid": 2,
        "value": "hi"
      }
    ]
  }
}
```

串口rpc请求包：

```

      | siid | eiid | piid | value | piid | value |
event_occured      1      1      1      17      2      "hi"
-----
```

串口命令执行event\_occured

步骤	串口命令
1	↑ event_occured 1 1 1 17 2 "hi"
2	↓ ok
或	
2	↓ error

code 状态码定义

0	成功	
1	接收到请求，但操作还没有完成	
-4001	属性不可读	
-4002	属性不可写	
-4003	属性、方法、事件不存在	
-4004	其他内部错误	
-4005	属性 value错误	
-4006	方法 in 参数错误	
-4007	did 错误	

待持续增加