

# 接口文档 1.1

2016 年 4 月 25 日

## 目录

<b>1</b>	<b>数据分类</b>	<b>2</b>
<b>2</b>	<b>action</b>	<b>2</b>
2.1	acquire . . . . .	2
2.2	release . . . . .	2
2.3	authorize . . . . .	3
2.4	broadcast . . . . .	3
<b>3</b>	<b>status</b>	<b>3</b>
3.1	authorized . . . . .	3
3.2	auth_failed . . . . .	4
3.3	key_pressed . . . . .	4
3.4	switch_on . . . . .	5
3.5	switch_off . . . . .	5
3.6	button_pressed . . . . .	5
3.7	button_released . . . . .	6
3.8	file_uploaded . . . . .	6
3.9	bit_file_programmed . . . . .	7
3.10	disk_file_downloaded . . . . .	7
<b>4</b>	<b>operation</b>	<b>7</b>
4.1	1 (键盘按键) . . . . .	7
4.2	2 (打开开关) . . . . .	7
4.3	3 (关闭开关) . . . . .	8
4.4	4 (按下按钮) . . . . .	8
4.5	5 (松开按钮) . . . . .	8
<b>5</b>	<b>info</b>	<b>9</b>
5.1	user_changed . . . . .	9
5.2	fpga_disconnected . . . . .	9
5.3	broadcast . . . . .	9

接口指服务器端、树莓派端与浏览器端之间进行通信的协议。三者间通过 WebSocket 技术进行双向通信，其中的数据通过 json 对象进行传递，具体规定如下。

## 1 数据分类

传输的数据分为四类：

1. 对服务器发起的动作信息，例如申请可用 FPGA、树莓派申请认证等。该类信息中主要字段为 **action**。
2. 对树莓派发起的操作信息，例如打开开关、按下按钮等。该类信息中主要字段为 **operation**。
3. 对于发起的动作返回的信息，例如确认上传成功、文件烧录失败等。该类信息中主要字段为 **status**。
4. 由于状态变化而给出的信息，例如树莓派连接断开等。该类信息中主要字段为 **info**。

因此所有数据包中有 **type** 字段，用于区分当前数据包的类型。其中对应关系如下：

```
{
  0: "action",
  1: "status",
  2: "operation",
  3: "info"
}
```

## 2 action

### 2.1 acquire

功能：用于申请可用的 FPGA 设备。

说明：由浏览器向服务器发出。

返回：服务器会返回 **user\_changed** 作为反馈，同时会将该信息广播给所有连接的服务器。

示例：

```
{
  type: 0,
  action: "acquire"
}
```

### 2.2 release

功能：用于释放当前控制的 FPGA 设备。

说明：由浏览器向服务器发出。

返回：服务器会返回 **user\_changed** 作为反馈，同时会将该信息广播给所有连接的服务器。

示例：

```
{
  type: 0,
  action: "release"
}
```

## 2.3 authorize

**功能：**树莓派申请进行身份验证。

**说明：**由树莓派向服务器发出。其中 `auth_key` 字段可通过 API 进行查询。

**返回：**

**示例：**

```
{
  type: 0,
  action: "authorize",
  device_id: "fpga",
  auth_key: "xxxxxxxxx"
}
```

## 2.4 broadcast

**功能：**用于请求服务器将消息进行转发。

**说明：**由浏览器向服务器发出。

**返回：**无

**示例：**

```
{
  type: 0,
  action: "broadcast",
  content: "xxxxxxxxx"
}
```

# 3 status

## 3.1 authorized

**功能：**用于表明树莓派验证通过。

**说明：**由服务器向树莓派发出。同时返回的数据中还会有以下字段：

- `index`: 当前树莓派分配的索引号
- `filelink`: 树莓派下载文件时所用的连接

- **webport**: 服务器提供下载文件所使用的 web 端口号
- **rtmp\_host**: rtmp 服务器地址
- **rtmp\_push\_port**: rtmp 视频流推送端口
- **stream\_name**: 视频流的名称

返回: 无。

示例:

```
{
  type: 1,
  status: "authorized",
  index: 0,
  filelink: "/live/0/file",
  webport: 9000,
  rtmp_host: "127.0.0.1",
  rtmp_push_port: 1935,
  stream_name: "0"
}
```

### 3.2 auth\_failed

**功能**: 用于表明树莓派验证失败。

**说明**: 由服务器向树莓派发出。

返回: 无。

示例:

```
{
  type: 1,
  status: "auth_failed"
}
```

### 3.3 key\_pressed

**功能**: 用于表明树莓派成功输入按键信号。

**说明**: 由树莓派向浏览器发出。该信息会被转发给所有观看该设备的浏览器页面。数据包还含有 **key\_code** 字段用于标示所按下的按键。

返回: 无。

示例:

```
{
  type: 1,
```

```
    status: "key_pressed",
    key_code: 20
}
```

### 3.4 switch\_on

**功能：**用于表明树莓派成功输入打开开关的信号。

**说明：**由树莓派向浏览器发出。该信息会被转发给所有观看该设备的浏览器页面。数据包还含有 `id` 字段用于标示开关的编号。

**返回：**无。

**示例：**

```
{
  type: 1,
  status: "switch_on",
  id: 1
}
```

### 3.5 switch\_off

**功能：**用于表明树莓派成功输入关闭开关的信号。

**说明：**由树莓派向浏览器发出。该信息会被转发给所有观看该设备的浏览器页面。数据包还含有 `id` 字段用于标示开关的编号。

**返回：**无。

**示例：**

```
{
  type: 1,
  status: "switch_off",
  id: 1
}
```

### 3.6 button\_pressed

**功能：**用于表明树莓派成功输入按下按钮的信号。

**说明：**由树莓派向浏览器发出。该信息会被转发给所有观看该设备的浏览器页面。数据包还含有 `id` 字段用于标示按钮的编号。

**返回：**无。

**示例：**

```
{
  type: 1,
```

```
    status: "button_pressed",
    id: 1
}
```

### 3.7 button\_released

**功能：**用于表明树莓派成功输入松开按钮的信号。

**说明：**由树莓派向浏览器发出。该信息会被转发给所有观看该设备的浏览器页面。数据包还含有 `id` 字段用于标示按钮的编号。

**返回：**无。

**示例：**

```
{
  type: 1,
  status: "button_released",
  id: 1
}
```

### 3.8 file\_uploaded

**功能：**用于表示 bit 文件上传成功。

**说明：**由服务器向浏览器和服务器发出。数据包还含有名为 `file` 的 json 对象，其中有以下字段：

- `type`: 表明当前文件的类型，可能的取值为 `bit` 和 `disk`
- `name`: 代表上传的文件名
- `size`: 代表上传文件的大小

**返回：**无。

**示例：**

```
{
  type: 1,
  status: "bit_file_uploaded",
  file:
  {
    type: "bit",
    name: "file.bit",
    size: 1000
  }
}
```

### 3.9 bit\_file\_programmed

功能：用于表示 bit 文件下载成功。

说明：由树莓派向浏览器发出。

返回：无。

示例：

```
{
  type: 1,
  status: "bit_file_programmed"
}
```

### 3.10 disk\_file\_downloaded

功能：用于表示树莓派下载 disk 文件成功。

说明：由树莓派向浏览器发出。

返回：无。

示例：

```
{
  type: 1,
  status: "disk_file_downloaded"
}
```

## 4 operation

### 4.1 1 (键盘按键)

功能：用于模拟键盘被按下。

说明：由浏览器向树莓派发出。数据包还含有 `key_code` 字段用于标示所按下的按键。

返回：树莓派会返回 `key_pressed` 作为确认。

示例：

```
{
  type: 2,
  operation: 1,
  key_code: 20
}
```

### 4.2 2 (打开开关)

功能：用于模拟开关被打开。

说明：由浏览器向树莓派发出。数据包还含有 `id` 字段用于标示开关的编号。

返回：树莓派会返回switch\_on作为确认。

示例：

```
{
  type: 2,
  operation: 2,
  id: 1
}
```

### 4.3 3（关闭开关）

功能：用于模拟开关被关闭。

说明：由浏览器向树莓派发出。数据包还含有 id 字段用于标示开关的编号。

返回：树莓派会返回switch\_off作为确认。

示例：

```
{
  type: 2,
  operation: 3,
  id: 1
}
```

### 4.4 4（按下按钮）

功能：用于模拟按钮被按下。

说明：由浏览器向树莓派发出。数据包还含有 id 字段用于标示开关的编号。

返回：树莓派会返回button\_pressed作为确认。

示例：

```
{
  type: 2,
  operation: 4,
  id: 1
}
```

### 4.5 5（松开按钮）

功能：用于模拟按钮被松开

说明：由浏览器向树莓派发出。数据包还含有 id 字段用于标示开关的编号。

返回：树莓派会返回button\_released作为确认。

示例：



```
{
  type: 2,
  operation: 5,
  id: 1
}
```

## 5 info

### 5.1 user\_changed

**功能：**用于表示树莓派的操作者发生了变更。

**说明：**由服务器向浏览器发出，同时也会发送给树莓派。数据包还含有 `user` 字段用于标示当前的使用者，该字段可为空 (`null`)，表示当前无人进行操作。

**返回：**

**示例：**

```
{
  type: 3,
  user: null
}
```

### 5.2 fpga\_disconnected

**功能：**用于表示树莓派的连接断开。

**说明：**由服务器向浏览器发出。

**返回：**无。

**示例：**

```
{
  type: 3,
  info: "fpga_disconnected"
}
```

### 5.3 broadcast

**功能：**用于广播服务器接收到的消息。

**说明：**由服务器向浏览器发出。

**返回：**无

**示例：**

```
{
  type: 3,
```

```
info: "broadcast",  
timestamp: "1461564393",  
nickname: "name",  
content: "xxxxxxx"  
}
```