

# AtebJS 使用手册

V3.5.20241120

此文档为上海敏拂计算机科技有限公司版权所有，未经许可，任何人不得以任何形式予以传播。涉及到的方案、图片、价格等均属上海敏拂计算机科技有限公司商业秘密，用户只能在公司内采用报告内之内容，不得将方案涉及内容对外公布。

# 目 录

第一章 概述 .....	3
1.1.简介 .....	3
1.2.初始化 .....	3
第二章 方法库说明 .....	4
2.1.Scanner .....	4
2.1.1.scannerData .....	4
2.2.Print .....	4
2.2.1.printData .....	4
2.3.GPIO .....	5
2.3.1.inputStatus .....	5
2.3.2.setOutput .....	5
2.3.3.SRNDFI044 .....	6
2.3.3.1.inputStatus .....	6
2.3.3.2.setOutput .....	7
2.3.3.3.init .....	7
2.4.SerialPort .....	8
2.4.1.openPort .....	8
2.4.2.serialPortData .....	8
2.4.3.transportData .....	9
2.4.4.closePort .....	10
2.4.5.endSymbol .....	11
2.4.6.getSerialList .....	12
2.4.7.refreshSerialList .....	12
2.5.DeviceSetting .....	13
2.5.1.openSettings .....	13
2.5.2.setVolume .....	14
2.6.RFID .....	15
2.6.1.open .....	15
2.6.2.getData .....	15
2.6.3.close .....	16

2.7.USB .....	17
2.7.1.zebra .....	17
2.7.1.1.findDevice .....	17
2.7.1.2.getMessage .....	18
2.7.1.3.send .....	19
2.8.OnMessage .....	20
2.8.1.onMessage .....	20

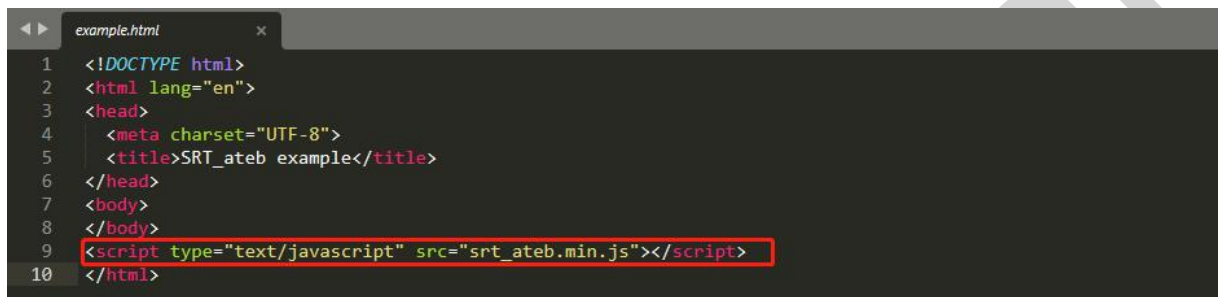
Confidential

## 第一章 概述

### 1.1.简介

AtebJS 是 SDCT 和 PDA 移动设备产品的 js 开发库。

使用时需在 SDCT 或 PDA 环境下使用,并将 srt\_atéb.min.js 引入到需要使用该文件的\*.html 代码中即可。



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>SRT_atéb example</title>
6 </head>
7 <body>
8 </body>
9 <script type="text/javascript" src="srt_atéb.min.js"></script>
10 </html>
```

### 1.2.初始化

在引入 srt\_atéb.min.js 之后,需要对 srt\_atéb 进行初始化后,才能调用 srt\_atéb 的方法,初始化的方法是 srt\_atéb.init();或者 srt\_atéb.init({});



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>SRT_atéb example</title>
6 </head>
7 <body>
8 </body>
9 <script type="text/javascript" src="srt_atéb.min.js"></script>
10 <script type="text/javascript">
11   srt_atéb.init(); // srt_atéb.init({});
12 </script>
13 </html>
```

## 第二章 方法库说明

### 2.1.Scanner

扫描器模块 (适用于 PDA 设备)

#### 2.1.1.scannerData

接收扫描信息，在初始化的时候传递回调函数，函数接收一个参数（扫描结果）。

```
example.html x
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>SRT_ateb example</title>
6 </head>
7 <body>
8 </body>
9 <script type="text/javascript" src="srt_ateb.min.js"></script>
10 <script type="text/javascript">
11   srt_ateb.init({
12     Scanner: {
13       scannerData: function(res) {
14         // ... body
15       }
16     }
17   });
18 </script>
19 </html>
```

### 2.2.Print

蓝牙打印机模块 (适用于 PDA 设备)

#### 2.2.1.printData

发送打印数据，在初始化完成之后，使用 `srt_ateb.Print.printData(mac, data)`调用该方法。其中 `mac` 是打印机的蓝牙 `mac` 地址，`data` 是需要打印的内容。

```
example.html x
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>SRT_ateb example</title>
6 </head>
7 <body>
8   <button onclick="printData()">print</button>
9 </body>
10 <script type="text/javascript" src="srt_ateb.min.js"></script>
11 <script type="text/javascript">
12   srt_ateb.init();
13   function printData() {
14     srt_ateb.Print.printData(mac, data);
15   }
16 </script>
17 </html>
```

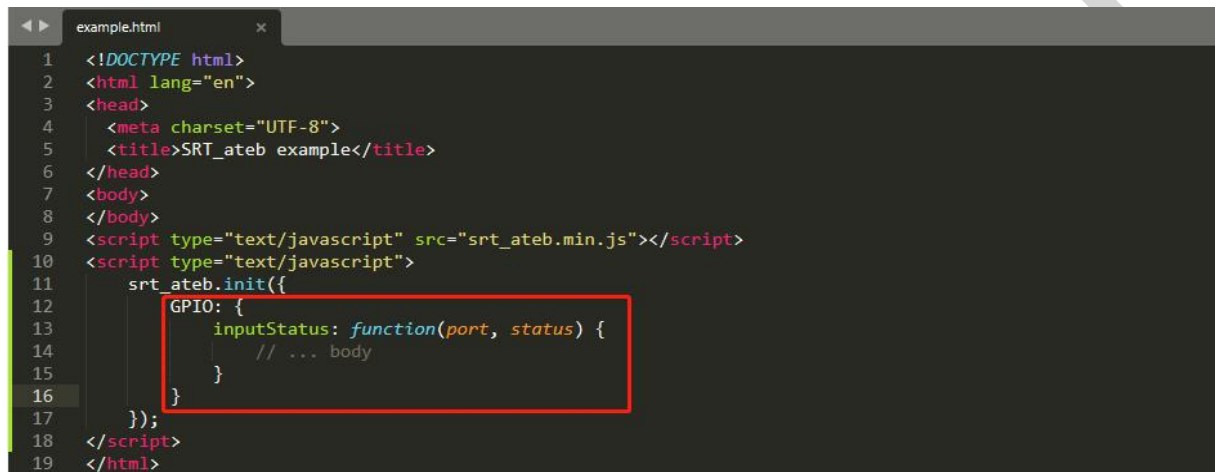
## 2.3.GPIO

输入输出模块 (适用于 SDCT 设备)

### 2.3.1.inputStatus

接收状态, 在初始化的时候传递回调函数, 函数接收两个参数 (端口, 端口状态)。

其中端口目前包含两个值 1 和 2, 状态分别是 0(开启), 1(关闭)

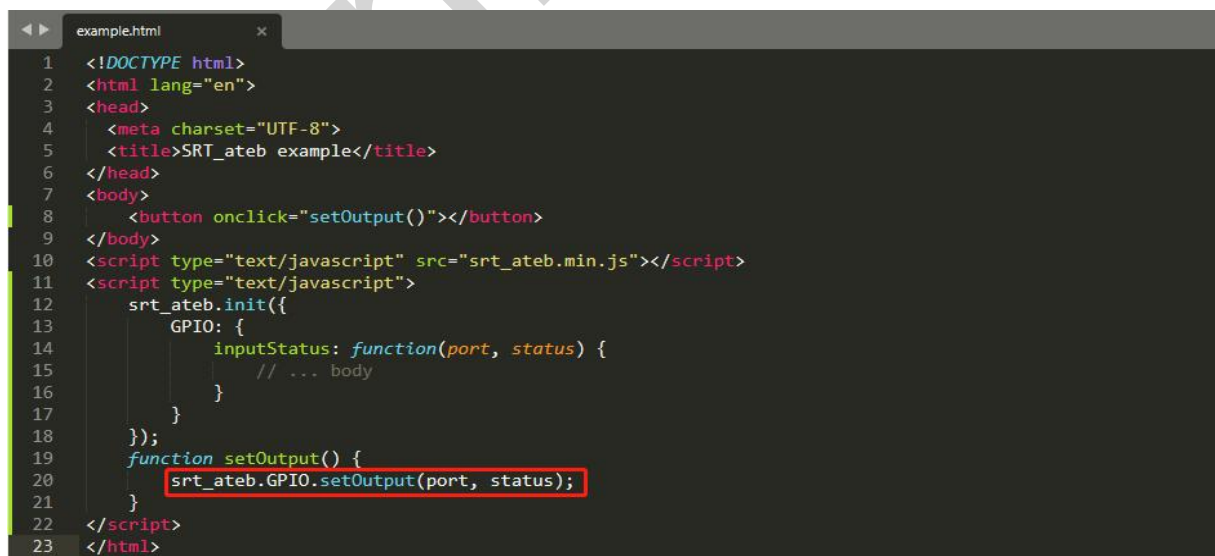


```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>SRT_ateb example</title>
6 </head>
7 <body>
8 </body>
9 <script type="text/javascript" src="srt_ateb.min.js"></script>
10 <script type="text/javascript">
11   srt_ateb.init({
12     GPIO: {
13       inputStatus: function(port, status) {
14         // ... body
15       }
16     }
17   });
18 </script>
19 </html>
```

### 2.3.2.setOutput

发送状态, 在初始化完成之后, 使用 srt\_ateb.GPIO.setOutput(port, status)调用该方法。

其中 port 是端口(1 和 2), status 是端口状态(1:开启, 0:关闭)。



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>SRT_ateb example</title>
6 </head>
7 <body>
8   <button onclick="setOutput()"></button>
9 </body>
10 <script type="text/javascript" src="srt_ateb.min.js"></script>
11 <script type="text/javascript">
12   srt_ateb.init({
13     GPIO: {
14       inputStatus: function(port, status) {
15         // ... body
16       }
17     }
18   });
19   function setOutput() {
20     srt_ateb.GPIO.setOutput(port, status);
21   }
22 </script>
23 </html>
```

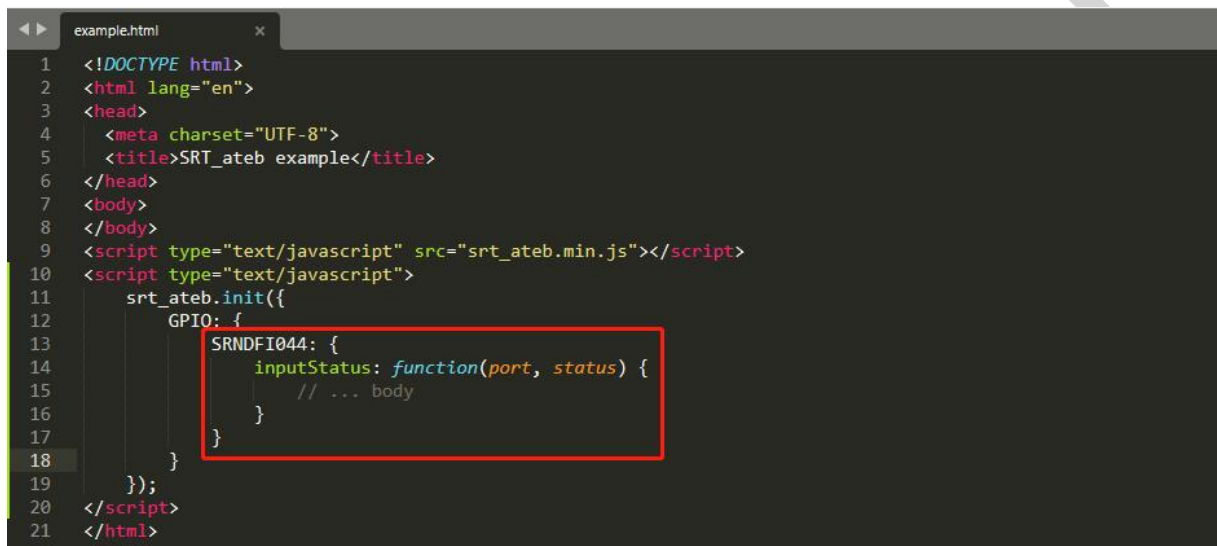
### 2.3.3.SRNDFI044

srt\_ateb.GPIO 控制 SRNDFI044 GPIO 的模块。

#### 2.3.3.1.inputStatus

接收状态，使用方法和 GPIO.inputStatus 相似，

参数 Port: 1~4(8), status: 0 (开启) 1 (关闭)



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>SRT_ateb example</title>
6 </head>
7 <body>
8 </body>
9 <script type="text/javascript" src="srt_ateb.min.js"></script>
10 <script type="text/javascript">
11   srt_ateb.init({
12     GPIO: {
13       SRNDFI044: {
14         inputStatus: function(port, status) {
15           // ... body
16         }
17       }
18     }
19   });
20 </script>
21 </html>
```

### 2.3.3.2.setOutput

发送状态，使用方法和 GPIO.setOutput 相似

参数 Port: 0~3(7), status: 0 (关闭) 1 (开启)

```
example.html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>SRT_ateb example</title>
6 </head>
7 <body>
8   <button onclick="setOutput()"></button>
9 </body>
10 <script type="text/javascript" src="srt_ateb.min.js"></script>
11 <script type="text/javascript">
12   srt_ateb.init({
13     GPIO: {
14       SRNDFI044: {
15         inputStatus: function(port, status) {
16           // ... body
17         }
18       }
19     }
20   });
21   function setOutput() {
22     srt_ateb.GPIO.SRNDFI044.setOutput(port, status);
23   }
24 </script>
25 </html>
```

### 2.3.3.3.init

对 SRNDFI044 的 DO 口进行初始化, 该参数接收一个布尔变量, false: 关闭所有的 DO 口,

true: 打开所有 DO 口。

```
example.html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>SRT_ateb example</title>
6 </head>
7 <body>
8   <button onclick="setOutput()">output</button>
9 </body>
10 <script type="text/javascript" src="srt_ateb.min.js"></script>
11 <script type="text/javascript">
12   srt_ateb.init({
13     GPIO: {
14       SRNDFI044: {
15         inputStatus: function(port, status) {
16           // ... body
17         }
18         init: true
19       }
20     }
21   });
22   function setOutput() {
23     srt_ateb.GPIO.SRNDFI044.setOutput(port, status);
24   }
25 </script>
26 </html>
```



## 2.4.SerialPort

串口模块 (适用于 SDCT 设备)

### 2.4.1.openPort

开启串口, 在初始化完成之后, 使用 `srt_ateb.SerialPort.openPort(port[串口名称], baudrate[波特率], databit[数据位], parity[校验位], stopbit[停止位], flowcontrol[流量控制])`调用该方法。

例: 开启名为 “serial1” 的 port。

`srt_ateb.SerialPort.openPort(“serial1”, 9600, 8, “None”, 1, “None”);`

```
example.html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>SRT_ateb example</title>
6 </head>
7 <body>
8   <button onclick="openPort()">open</button>
9 </body>
10 <script type="text/javascript" src="srt_ateb.min.js"></script>
11 <script type="text/javascript">
12   srt_ateb.init();
13   function openPort() {
14     srt_ateb.SerialPort.openPort(port, baudrate, databit, parity, stopbit, flowcontrol);
15   }
16 </script>
17 </html>
```

### 2.4.2.serialPortData

接收串口消息, 在初始化的时候传递回调函数, 函数接收两个参数 (串口名称, 接收的数据)。

```
example.html
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <title>Document</title>
7 </head>
8
9 <body>
10  <button onclick="openPort()">open</button>
11 </body>
12 <script type="text/javascript" src="srt_ateb.min.js"></script>
13 <script type="text/javascript">
14   window.onload = function() {
15     srt_ateb.init({
16       SerialPort: {
17         serialPortData: function(port, data) {
18           // ... body
19         }
20       }
21     });
22     function openPort() {
23       srt_ateb.SerialPort.openPort(port, baudrate, databit, parity, stopbit, flowcontrol);
24     }
25   }
26 </script>
27
28 </html>
```

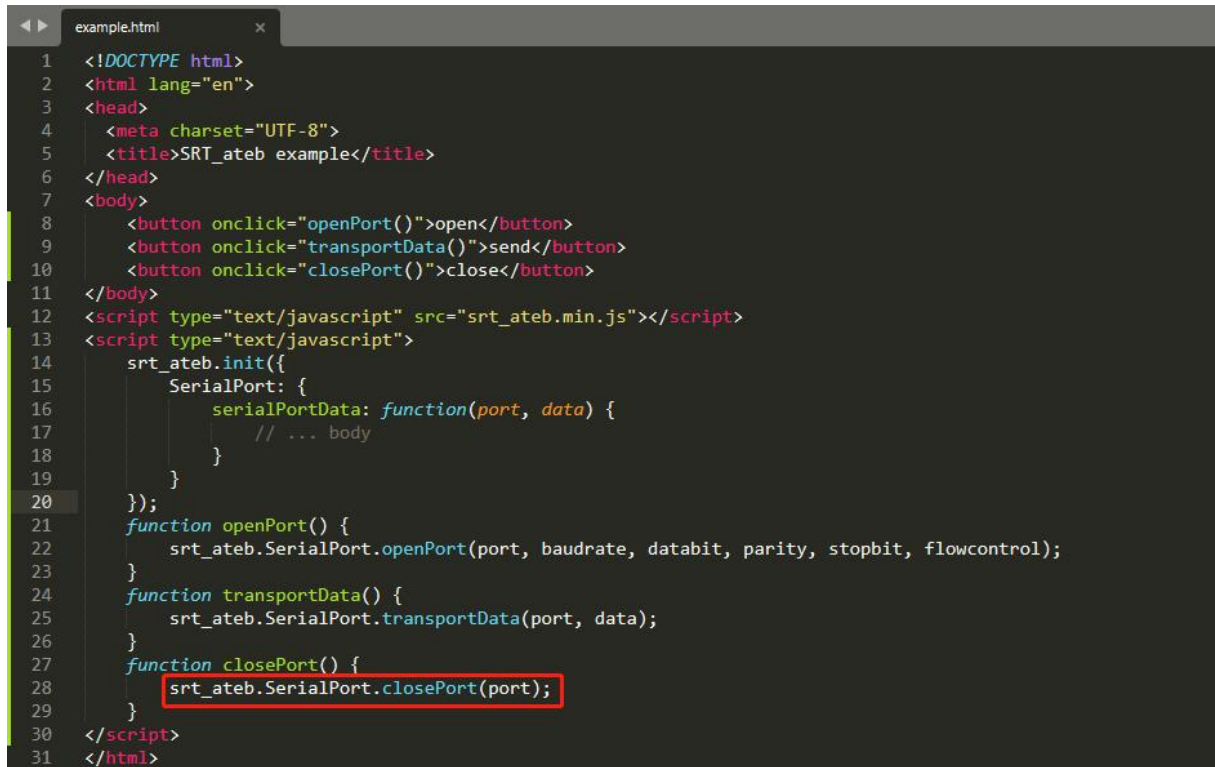
### 2.4.3.transportData

发送串口消息，在初始化完成之后，使用 `srt_atob.SerialPort.transportData(port, data)`调用该方法，其中 `port` 是串口名称，`data` 是发送的数据(`data` 默认发送的是十六进制的数据，发送的时候请注意数据的类型转换)。

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="initial-scale=1.0, maximum-scale=1.0, user-scalable=no" />
6     <title>SRT_atob example</title>
7   </head>
8   <body>
9     <button onclick="transportData()">send</button>
10  </body>
11  <script src="srt_atob.min.js" type="text/javascript" charset="utf-8"></script>
12  <script type="text/javascript">
13    srt_atob.init({
14      SerialPort: {
15        serialPortData: function(port, data) {
16          console.log(port, data);
17        }
18      }
19    });
20
21    function transportData() {
22      srt_atob.SerialPort.transportData(port, data);
23    }
24  </script>
25 </html>
```

#### 2.4.4.closePort

关闭串口的方法，在初始化完成之后，使用 `srt_ateb.SerialPort.closePort(port)`调用该方法，参数 `port` 是串口的名称。



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>SRT_ateb example</title>
6 </head>
7 <body>
8   <button onclick="openPort()">open</button>
9   <button onclick="transportData()">send</button>
10  <button onclick="closePort()">close</button>
11 </body>
12 <script type="text/javascript" src="srt_ateb.min.js"></script>
13 <script type="text/javascript">
14   srt_ateb.init({
15     SerialPort: {
16       serialPortData: function(port, data) {
17         // ... body
18       }
19     }
20   });
21   function openPort() {
22     srt_ateb.SerialPort.openPort(port, baudrate, databit, parity, stopbit, flowcontrol);
23   }
24   function transportData() {
25     srt_ateb.SerialPort.transportData(port, data);
26   }
27   function closePort() {
28     srt_ateb.SerialPort.closePort(port);
29   }
30 </script>
31 </html>
```

### 2.4.5.endSymbol

串口数据的结束符，需要在初始化的时候设定（可不传），接收的是一个内容为十六进制的字符串（原因是为了可以使用一些十进制无法输入的特殊字符），串口在接收到数据的时候，会对数据进行拼接，并在使用传入的参数分割数据之后，依次发出，因此会存在一些延迟；在不进行设置的时候，默认不进行任何操作，将数据直接发出。

```
example.html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <title>SRT_ateb example</title>
6  </head>
7  <body>
8    <button onclick="openPort()">open</button>
9    <button onclick="transportData()">send</button>
10   <button onclick="closePort()">close</button>
11 </body>
12 <script type="text/javascript" src="srt_ateb.min.js"></script>
13 <script type="text/javascript">
14   srt_ateb.init({
15     SerialPort:{
16       serialPortData:function(port, data){}
17     },
18     endSymbol: '0D 0A' // 数据结束符
19   });
20   function openPort() { // 打开串口
21     srt_ateb.SerialPort.openPort(port, baudrate, databit, parity, stopbit, flowcontrol);
22   }
23   function transportData() { // 发送数据
24     srt_ateb.SerialPort.transportData(port, data);
25   }
26   function closePort() { // 关闭串口
27     srt_ateb.SerialPort.closePort(port);
28   }
29 </script>
30 </html>
```

#### 2.4.6.getSerialList

获取当前可用的串口列表和描述，参数为一个 json 字符串。

例：[{“name”:“port1”,“description”:“串口 1”}]

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <meta name="viewport" content="initial-scale=1.0, maximum-scale=1.0, user-scalable=no" />
6   <title>SRT_ateb example</title>
7 </head>
8 <body></body>
9 <script src="srt_ateb.min.js" type="text/javascript" charset="utf-8"></script>
10 <script type="text/javascript">
11   srt_ateb.init({
12     SerialPort: {
13       getSerialList: function(data) {
14         let res = JSON.parse(data);
15         console.log(res);
16       }
17     }
18   });
19 </script>
20 </html>
```

#### 2.4.7.refreshSerialList

刷新 getSerialList 获取的串口列表

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <meta name="viewport" content="initial-scale=1.0, maximum-scale=1.0, user-scalable=no" />
6   <title>SRT_ateb example</title>
7 </head>
8 <body>
9   <button onclick="refreshSerialList()">refresh</button>
10 </body>
11 <script src="srt_ateb.min.js" type="text/javascript" charset="utf-8"></script>
12 <script type="text/javascript">
13   srt_ateb.init({
14     SerialPort: {
15       getSerialList: function(data) {
16         let res = JSON.parse(data);
17         console.log(res);
18       }
19     }
20   });
21
22   function refreshSerialList() {
23     srt_ateb.SerialPort.refreshSerialList();
24   }
25 </script>
26 </html>
```



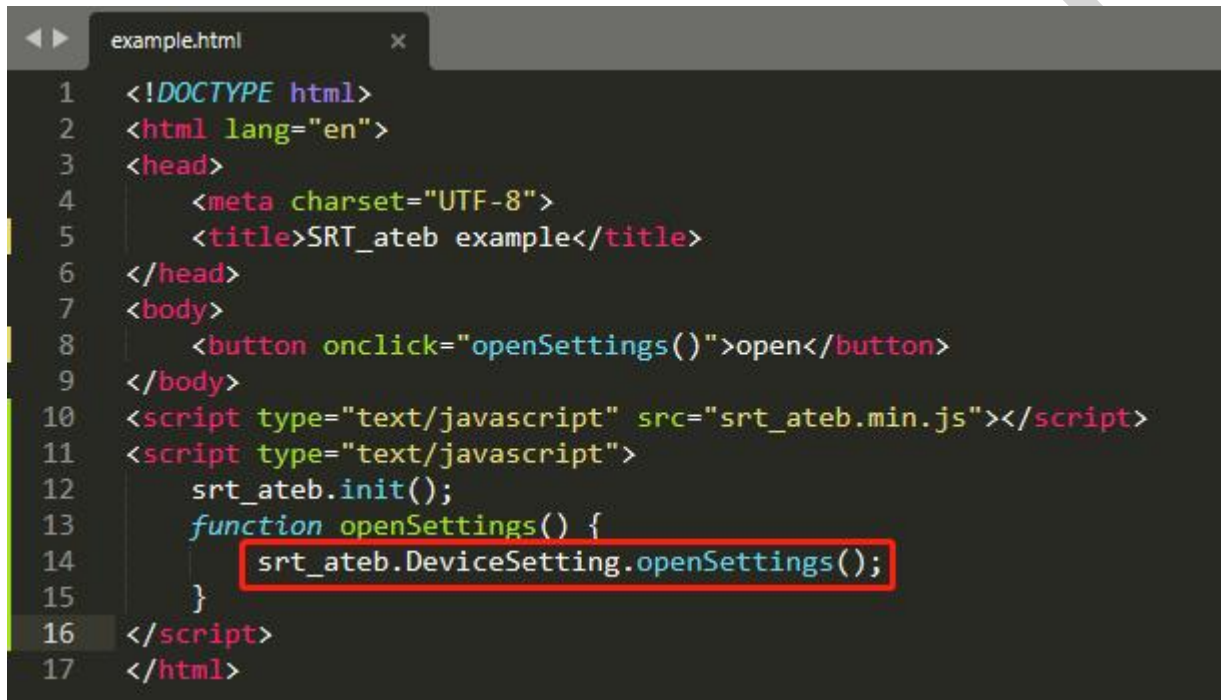
## 2.5.DeviceSetting

设备设置 (适用于 SDCT 设备)

### 2.5.1.openSettings

调出设定画面的方法，在初始化完成之后，使用 `srt_ateb.DeviceSetting.openSettings()`;

不需要参数。



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>SRT_ateb example</title>
6 </head>
7 <body>
8   <button onclick="openSettings()">open</button>
9 </body>
10 <script type="text/javascript" src="srt_ateb.min.js"></script>
11 <script type="text/javascript">
12   srt_ateb.init();
13   function openSettings() {
14     srt_ateb.DeviceSetting.openSettings();
15   }
16 </script>
17 </html>
```

### 2.5.2.setVolume

设置音量的方法，在初始化完成之后，使用 `srt_atvb.DeviceSetting.setVolume(type, volume)`，其中 `type` 是设置的类型(包括['ring(铃声)', 'music(音乐)', 'alarm(通知)', 'all(全部，设置统一的音量大小)', 'mute(静音模式)']，当 `type` 为 `mute` 时，`volume` 只有 1（打开静音），0（关闭静音）两个参数)，`volume` 是音量值，范围 0~15。

```
example.html x
7 <body>
8   <button onclick="openSettings()">open</button>
9   <select id="type">
10     <option value="ring" selected="true">RING</option>
11     <option value="music">MUSIC</option>
12     <option value="alarm">ALARM</option>
13     <option value="all">ALL</option>
14   </select>
15   <input type="text" id="volume" placeholder="音量值">
16   <button onclick="setVolume()">setVolume</button>
17   <button onclick="mute()">MUTE</button>
18   <button onclick="unmute()">UNMUTE</button>
19 </body>
20 <script type="text/javascript" src="srt_atvb.min.js"></script>
21 <script type="text/javascript">
22   srt_atvb.init();
23   function openSettings() {
24     srt_atvb.DeviceSetting.openSettings();
25   }
26   function setVolume() { // 设置音量
27     var type = document.getElementById('type').value;
28     var volume = document.getElementById('volume').value;
29     srt_atvb.DeviceSetting.setVolume(type, volume);
30   }
31   function mute() { // 打开静音
32     var type = 'mute';
33     var volume = 1;
34     srt_atvb.DeviceSetting.setVolume(type, volume);
35   }
36   function unmute() { // 关闭静音
37     var type = 'mute';
38     var volume = 0;
39     srt_atvb.DeviceSetting.setVolume(type, volume);
40   }
41 </script>
```

## 2.6.RFID

控制 RFID 扫描设备(适用于 PDA)

### 2.6.1.open

控制设备开始扫描 RFID 标签，在初始化完成之后，使用 `srt_ateb.RFID.open()`调用该方法，不需要参数。

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="initial-scale=1.0, maximum-scale=1.0, user-scalable=no" />
6     <title>SRT_ateb example</title>
7   </head>
8   <body>
9     <button onclick="open()">open</button>
10  </body>
11  <script src="srt_ateb.min.js" type="text/javascript" charset="utf-8"></script>
12  <script type="text/javascript">
13    srt_ateb.init();
14    function open() {
15      srt_ateb.RFID.open();
16    }
17  </script>
18 </html>
```

### 2.6.2.getData

获取设备扫描到的 RFID 数据，在初始化的时候传递回调函数，函数接收一个参数。

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="initial-scale=1.0, maximum-scale=1.0, user-scalable=no" />
6     <title>SRT_ateb example</title>
7   </head>
8   <body>
9     <button onclick="open()">open</button>
10  </body>
11  <script src="srt_ateb.min.js" type="text/javascript" charset="utf-8"></script>
12  <script type="text/javascript">
13    srt_ateb.init({
14      RFID: {
15        getData: function(data) {}
16      }
17    });
18    function open() {
19      srt_ateb.RFID.open();
20    }
21  </script>
22 </html>
```



### 2.6.3.close

控制设备结束扫描 RFID 标签，在初始化完成之后，使用 `srt_atvb.RFID.close()`调用该方法，不需要参数。

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="initial-scale=1.0, maximum-scale=1.0, user-scalable=no" />
6     <title>SRT_atvb example</title>
7   </head>
8   <body>
9     <button onclick="open()">open</button>
10    <button onclick="close()">close</button>
11  </body>
12  <script src="srt_atvb.min.js" type="text/javascript" charset="utf-8"></script>
13  <script type="text/javascript">
14    srt_atvb.init({
15      RFID: {
16        getData: function(data) {}
17      }
18    });
19    function open() {
20      srt_atvb.RFID.open();
21    }
22    function close() {
23      srt_atvb.RFID.close();
24    }
25  </script>
26 </html>
```

## 2.7.USB

通过设备携带的 USB 接口进行消息通信(适用于 SDCT)

### 2.7.1.zebra

srt\_ateb.USB 控制 zebra 的模块。

#### 2.7.1.1.findDevice

控制搜索可发现的 zebra 设备，在初始化完成之后，使用 srt\_ateb.USB.zebra.findDevice()调用该方法，不需要参数。

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <meta name="viewport" content="initial-scale=1.0, maximum-scale=1.0, user-scalable=no" />
6   <title>SRT_ateb example</title>
7 </head>
8 <body>
9   <button onclick="findDevice()">findDevice</button>
10 </body>
11 <script src="srt_ateb.min.js" type="text/javascript" charset="utf-8"></script>
12 <script type="text/javascript">
13   srt_ateb.init();
14   function findDevice() {
15     srt_ateb.USB.zebra.findDevice();
16   }
17 </script>
18 </html>
```

### 2.7.1.2.getMessage

获取设备搜索到的 zebra 信息,在初始化的时候传递回调函数,函数接收两个参数(type,list),其中 type 是接收消息的类型(目前包括‘findDevices<发现的设备>’),list 是接收消息的内容。

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <meta name="viewport" content="initial-scale=1.0, maximum-scale=1.0, user-scalable=no" />
6   <title>SRT_atéb example</title>
7 </head>
8 <body>
9   <button onclick="findDevice()">findDevice</button>
10 </body>
11 <script src="srt_atéb.min.js" type="text/javascript" charset="utf-8"></script>
12 <script type="text/javascript">
13   srt_atéb.init({
14     USB: {
15       zebra: {
16         getMessage: function(type, list) {}
17       }
18     }
19   });
20   function findDevice() {
21     srt_atéb.USB.zebra.findDevice();
22   }
23 </script>
24 </html>
```

### 2.7.1.3.send

控制向指定 zebra 设备发送消息，在初始化完成之后调用，使用时需要两个参数

`srt_atvb.USB.zebra.send(device, data)`, `device` 是目标设备的信息{"**ProductName**":"c 产品名称", "**ManufacturerName**":"厂商", "**SerialNumber**":"序列号", "**VendorId**":"供应商 ID (VID)", "**ProductId**":"产品识别码 (PID)", "**DeviceName**":"设备路径"} , `data` 是等待发送的信息。

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="utf-8">
5      <meta name="viewport" content="initial-scale=1.0, maximum-scale=1.0, user-scalable=no" />
6      <title>SRT_atvb example</title>
7  </head>
8  <body>
9      <button onclick="findDevice()">findDevice</button>
10     <button onclick="send()">send</button>
11 </body>
12 <script src="srt_atvb.min.js" type="text/javascript" charset="utf-8"></script>
13 <script type="text/javascript">
14     srt_atvb.init({
15         USB: {
16             zebra: {
17                 getMessage: function(type, list) {}
18             }
19         }
20     });
21     function findDevice() {
22         srt_atvb.USB.zebra.findDevice();
23     }
24     function send() {
25         var device; // 选择的已发现设备的信息
26         var data;   // 等待发送的信息
27         srt_atvb.USB.zebra.send(device, data);
28     }
29 </script>
30 </html>
```

## 2.8.OnMessage

消息回馈接口（目前支持串口信息）

### 2.8.1.onMessage

接收控制 ateb 进行操作后的结果反馈，返回内容是一个 JSON 格式的数据。

例：在调用 `srt_ateb.SerialPort.openPort()`后收到的回馈信息。

```
{  
  "object": "SerialPort",    // \\  
  "method": "openPort",     // → 调用目标  
  "port": "/dev/serialPort", // ↗  
  "code": 1000,              // → 回馈状态  
  "message": "Serial port /dev/serialPort successfully opened" // → 回馈结果  
}
```

```
1  <!DOCTYPE html>  
2  <html>  
3  <head>  
4    <meta charset="utf-8">  
5    <meta name="viewport" content="initial-scale=1.0, maximum-scale=1.0, user-scalable=no" />  
6    <title>SRT_ateb example</title>  
7  </head>  
8  <body>  
9  
10 </body>  
11 <script src="srt_ateb.min.js" type="text/javascript" charset="utf-8"></script>  
12 <script type="text/javascript">  
13   srt_ateb.init({  
14     OnMessage: {  
15       onMessage: function(data) {  
16         // TODO...  
17       }  
18     }  
19   });  
20 </script>  
21 </html>
```