 Jeremy_young 128

Serialport.js 连接 web 和硬件设备编程

javascript iot node.js 发布于 2017-10-20

Serialport 简介

想象这样一个世界，在那里你能用 JavaScript 代码控制榨汁机，灯，安防系统，甚至机器人。嗯，是机器人！你会不会觉得很新奇以致兴奋?

> Serialport 库（也称 Node-Serialport，基于 Node），为低级串口编码提供必要的 steam 接口，以控制 Arduino 芯片组，X10 接口，Zigbee 无线技术，公路路标，LCD 显示屏，收银抽屉，电机控制器，传感器，叉车，调制解调器，无人机，数控机床，绘图仪器，自动贩卖机，基于 ccTalk 协议的投币设备，SMS 网关，RFID 扫描器等等非常多设备。如果你手中有一块能够异步收发消息的硬件设备（我们姑且这样说），那么这个物理世界将成为你的掌中玩物。

Serialport 为 JavaScript 开发者打开了硬件开发之门。它是一个比编写固件更好的方案!

获取到 USB 串口路径

PC 机一般会带有 2 ~ 4 个 USB 插口， 以下称 port 口。不同的操作系统，获取到的串口信息不同。

欲了解 port 口信息，可以在命令行工具中输入命令：`serialport-list`。

Mac OSX 的 port 口为：

```
{
  comName: '/dev/tty.usbmodem1421',
  manufacturer: 'Arduino (www.arduino.cc)'
}
```

Linux 的 port 口为：

```
{
  comName: '/dev/ttyACM0',
  manufacturer: 'Arduino (www.arduino.cc)'
}
```

Windows 的 port 口为：

```
{
  comName: 'COM3',
  manufacturer: 'Arduino LLC (www.arduino.cc)'
}
```

其中， `comName` 字段，指的就是 USB 串口的路径。该路径是 `SerialPort` 实例化的依据。

获取串口列表：`SerialPort.list([callback]) ⇒ Promise`

因历史版本的缘故，该接口支持两种形式调用，推荐 v6.0.0 版本的 promise 方式：

```
// v4.0.7 的 callback 形式
SerialPort.list((error, ports) => console.log(ports))

// v6.0.0 的 promise 形式
SerialPort.list().then(ports => console.log(ports))
```

创建一个 SerialPort 对象

创建 SerialPort 对象： `new SerialPort(path, [options], [openCallback])`

有了 port 口路径，就可以创建一个 port 口实例，并建立连接。

```
let port = new SerialPort('/dev/tty.usbmodem1421');
```

该实例化是首先产生一个 port 实例，然后再尝试建立连接的。即实例化过程中有一个异步操作，实例化完成了，连接的结果可能还没有返回。

连接建立成功，就会触发 `open` 事件——事件稍后再解说。

合并以上两步的代码，就是：

```
import SerialPort from serialport;

SerialPort.list().then( ports => {
    // 假设选择第一个串口实例化
    let path = ports[0].comName;
    let myPort = new SerialPort(path);
})
.catch(err => console.log(err))
```

绑定事件监听

当获取到了 `SerialPort` 的实例对象 `myPort` 后，就可以进行事件监听了。

```
// 当连接建立时
myPort.on('open', callback);

// 当接收到数据时
myPort.on('data', callback);

// 当出现错误时
myPort.on('error', callback);
```

事件监听，主要用来在合适的时间点发送数据，以及处理接收到来自串口的数据信息。

值得注意的是，很多错误来自：因串口路径不对导致的连接错误（但此时实例对象已存在）、串口被占用锁定时仍尝试连接的错误。

向串口写入数据

向串口写入数据： `serialPort.write(data, [encoding], [callback]) ⇒ boolean`

实例创建完，并且正确建立连接后，就可以向串口写数据了。数据会经串口发送至与 PC 连接的硬件设备，比如 Arduino 板，或者 Raspberry Pi 板等等。

```
// 直接写入字符串
myPort.write('hello world', (err) => {
    if (err) return console.log('write Error: ', err.message);
})

// 写入 Buffer
myPort.write(Buffer.from('hello world'), callback)
```

写入数据完毕，就会调用上述回调。

若写数据出错——可能因为数据非法或断开了连接等原因——同样会调用上述回调，只不过有些错误情况下，可能 `err` 参数不存在。但是 `error` 事件一定会被触发。

连接未建立，即 `open` 事件未被触发，若此前就写入数据，写操作会被阻塞，直到建立连接之后再执行。

串口每次传输数据是有一定长度限制的。一个数据包写完，才会开始写下一个数据包；若一条数据太长，会被切分成多个包，依次写入。写完后会立即调用 `drain` 方法表示本条数据已写完，`drain` 意为排干了拥塞的数据。

一些安装 trouble

这里主要是 `serialport` 一些安装不成功的问题，包括 Windows 系统，Electron （跨平台的框架），一些 Linux 发行版以及 Raspberry Pi 板，都有可能发生一些安装的麻烦。 难以一一呈现，需耐心 Google~

文末彩蛋

如开篇说说，Serialport 是基于 Node 的一个 JS 库，那么上述代码需要在 Node 环境中运行，也就是我们日常的命令行。但如果想直接在浏览器中使用，还有一段距离。

所以，为更好的服务于 web 开发，这里有一款本人封装的 npm 工具——sensorium-server，只需在命令行中开启此工具，就可以轻松搭建一个从 HTML 页面到硬件设备的连接通道，这样就可以在 Browser 中轻松的调试硬件了。

阅读 7.2k • 发布于 2017-10-20

Jeremy_young
128

关注作者

6 条评论

得票 •时间

撰写评论 ...

提交评论

滴有乔沐： 这个报错是什么意思啊
warning in ./node_modules/bindings/bindings.js
81:43-53 Critical dependency: the request of a dependency is an expression

• 回复 • 2018-04-09

Jeremy_young： 依赖包有bug? npm install bindings ?

• 回复 • 2018-05-15

Django： 前辈你封装的sensorium-server，可以设置波特率吗?

• 回复 • 2019-05-07

> **Jeremy_young**： 回复略晚。。不能，因为只是一个简单特定工具，之前为了满足内部开发做的。
>
> • 回复 • 2019-12-14

老冒儿： 你好 大佬向您请教个问题：electron-vue写的一个项目 在一个页面写了打开串口通信 也能收到发过来的数据 但是当我路由跳转再回到这个页面 页面上就不会显示 但是控制台是能打印出接收到的数据(就是serialPort.on("data", function(data)data打印是有数据的) 这个是什么原因呢?

• 回复 • 2019-12-04

> **Jeremy_young**： 看你的描述监听到的数据没有通知到转回的路由页面里，是不是监听这块出bug了？最好把监听通信的代码放在主逻辑里，不被会路由刷掉的地方。具体还靠你debug...
>
> • 回复 • 2019-12-14

推荐阅读

YodaOS：一个属于 Node.js 社区的操作系统

大家好，很开心在这里宣布 YodaOS 开源了。他将承载 Rokid 4年以来对于人工智能和语音交互领域的沉淀，并选择 Node.js 作为...

Rokid技术团队 • 阅读 5.3k • 55 赞 • 6 评论

JavaScript 就要统治世界了?

" JavaScript 可以……""嘛，不就是操作一下 DOM，可以让元素飞来飞去吗""JavaScript 是……""不就是用 jQuery 让网页动起来，顶...

XBT1 • 阅读 35.8k • 50 赞 • 26 评论

JavaScript深入浅出第4课：V8引擎是如何工作的?

摘要： 性能彪悍的V8引擎。 《JavaScript深入浅出》系列： JavaScript深入浅出第1课：箭头函数中的this究竟是什么鬼？ JavaScrip...

Fundebug • 阅读 1k • 16 赞

阿里云前端周刊 - 第 14 期

随着近年来 Web 的发展与 JavaScript 的崛起，JavaScript 被应用到了许多原本不曾想象到的场景中，从服务端、工作站、数据库、...

Houfeng • 阅读 3.8k • 6 赞

JavaScript在物联网中的应用

凡是能用JavaScript写出来的，最终都会用JavaScript写出来。 —— Atwood定律 在那篇《最流行的编程语言JavaScript能做什么？...

phodal • 阅读 3.7k • 6 赞

快速学习nodejs系列：三、nodejs是什么

nodejs官网（nodejs.org）上的定义：Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine. Node.js uses an ev...

小小心 • 阅读 1k • 6 赞

按步搭建简单IoT微服务(1)

本文翻译自nearform的微服务开发工作坊项目，网址如下：[链接]文章介绍了实现IoT微服务相关模块，可以按照例子（已经提供代...

puyu • 阅读 1.2k • 3 赞

翻译 | 摆脱浏览器限制的JavaScript

技术世界在发展，JavaScript也在同步发展。JavaScript在软件世界建起地盘的头几年，它从没想过涉足服务应用程序、移动端应用...

iKcamp • 阅读 2.1k • 2 赞 • 2 评论