**Wonton快速开发手册**

（为wonton添加你自定义的器件）

2021年9月

目录

[一、Wonton简介 2](#_Toc83770610)

[1.1wonton开发简介 2](#_Toc83770611)

[1.2阅读本文须知 3](#_Toc83770612)

[1.3wonton相关文献 3](#_Toc83770613)

[二、环境搭建 4](#_Toc83770614)

[2.1安装visual studio 4](#_Toc83770615)

[2.2开始以调试模式运行wonton 5](#_Toc83770616)

[三、添加器件 8](#_Toc83770617)

[3.1器件简介 8](#_Toc83770618)

[3.1.1 Input器件 8](#_Toc83770619)

[3.1.2 output器件 9](#_Toc83770620)

[3.2 如何添加你自定义的器件 10](#_Toc83770621)

[3.2.1 创建器件代码 10](#_Toc83770622)

[3.2.2在Devices.js中声明 11](#_Toc83770623)

[3.3 为你更新过后的wonton制作安装包 12](#_Toc83770624)

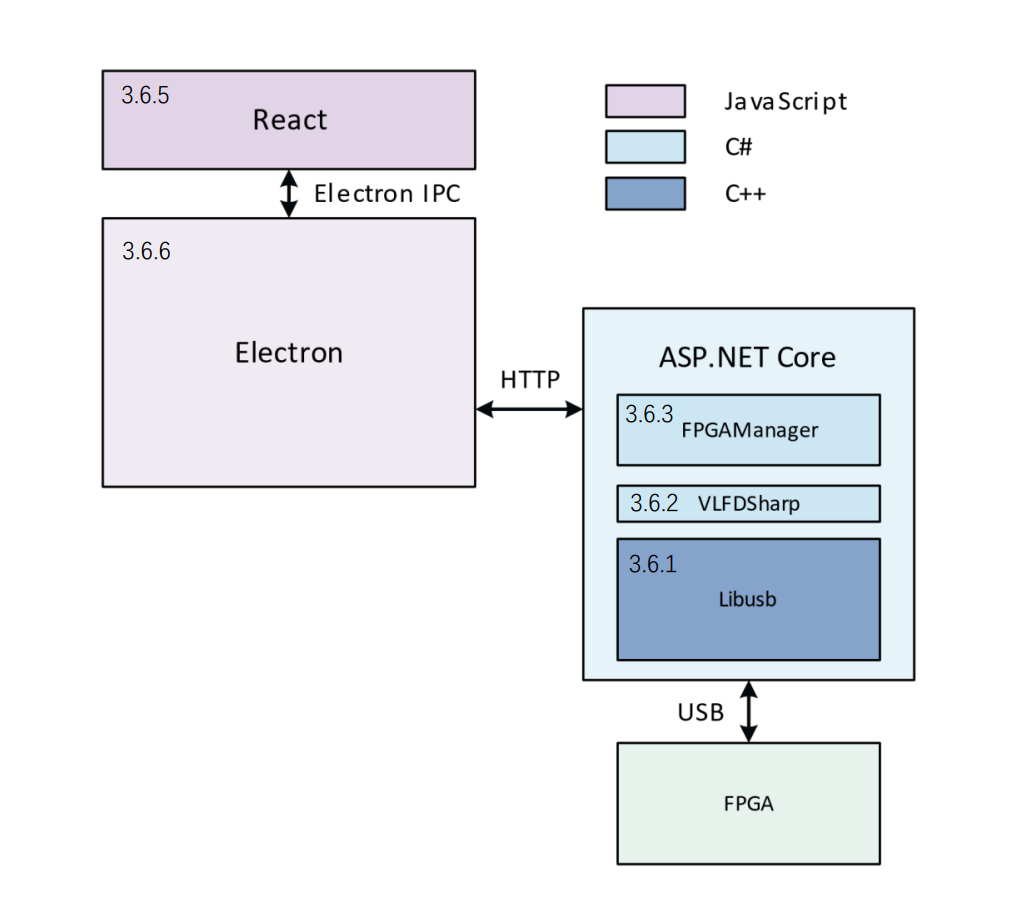
[四、其他 12](#_Toc83770625)

# 一、Wonton简介

## 1.1wonton开发简介

Wonton是一个开源的虚拟FPGA GUI软件。

其项目架构如下：



Wonton是一个前后端分离的MVC式架构。数据的处理核心是FPGA板，软件本身完成的是数据传输功能。后端基于C++，C#开发，直接与FPGA交互。前端基于electron+React的框架开发，与用户进行交互。前后端之间通过HTTP协议传输数据。

本文的目的在于让开发人员快速为wonton添加新的器件，因此开发人员只需要了解前端中一部分代码的架构。对有兴趣的同学，也欢迎到项目github主页做更深一步的学习。

## 1.2阅读本文须知

Wonton的前端实际上和web开发流程非常像，掌握一定前端开发知识的同学应该可以非常快速的上手。需要掌握的基础是javascript语言，以及对css/HTML有一定的了解，实际上后两者都非常的简单易上手，前者风格和C有些差异，但也是非常好上手的一门语言。

掌握了基础知识后，还需要掌握本项目使用的React库，React是目前前端开发比较热门的一个库，网上资料非常多。

掌握了这两个基本知识，就可以上手添加你的自定义器件了。没有熟练掌握也没关系，在现有代码基础上照着来也能快速上手。

比较花费时间的部分在于环境搭建，请参阅本文第二章。

## 1.3wonton相关文献

javascript/css/html推荐学习网站：<https://www.runoob.com/>

javascript函数查阅官网：[JavaScript | MDN (mozilla.org)](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript)

React官方文档：[开始 – React (docschina.org)](https://react.docschina.org/docs/getting-started.html)

主要使用的React器件库：<http://reactstrap.github.io/components>

Wonton目前维护网页：<https://github.com/Hi2129/Wonton_master>

Wonton目前的文档网页：<https://github.com/Hi2129/Wonton_doc>

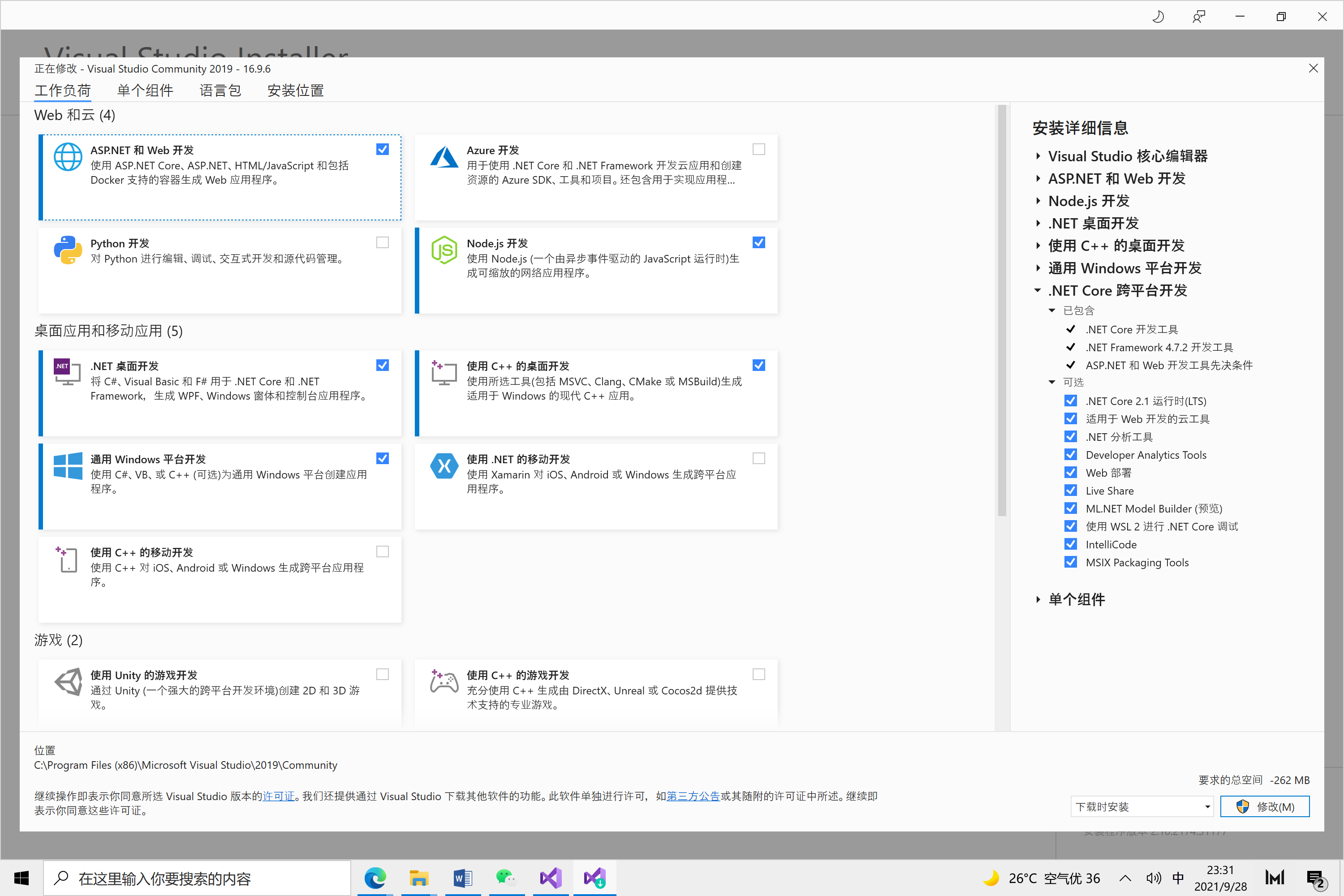
# 二、环境搭建

## 2.1安装visual studio

高度推荐在windows下搭建开发环境，电脑性能够好的可以开个虚拟机。

Visual studio的安装在此不赘述。

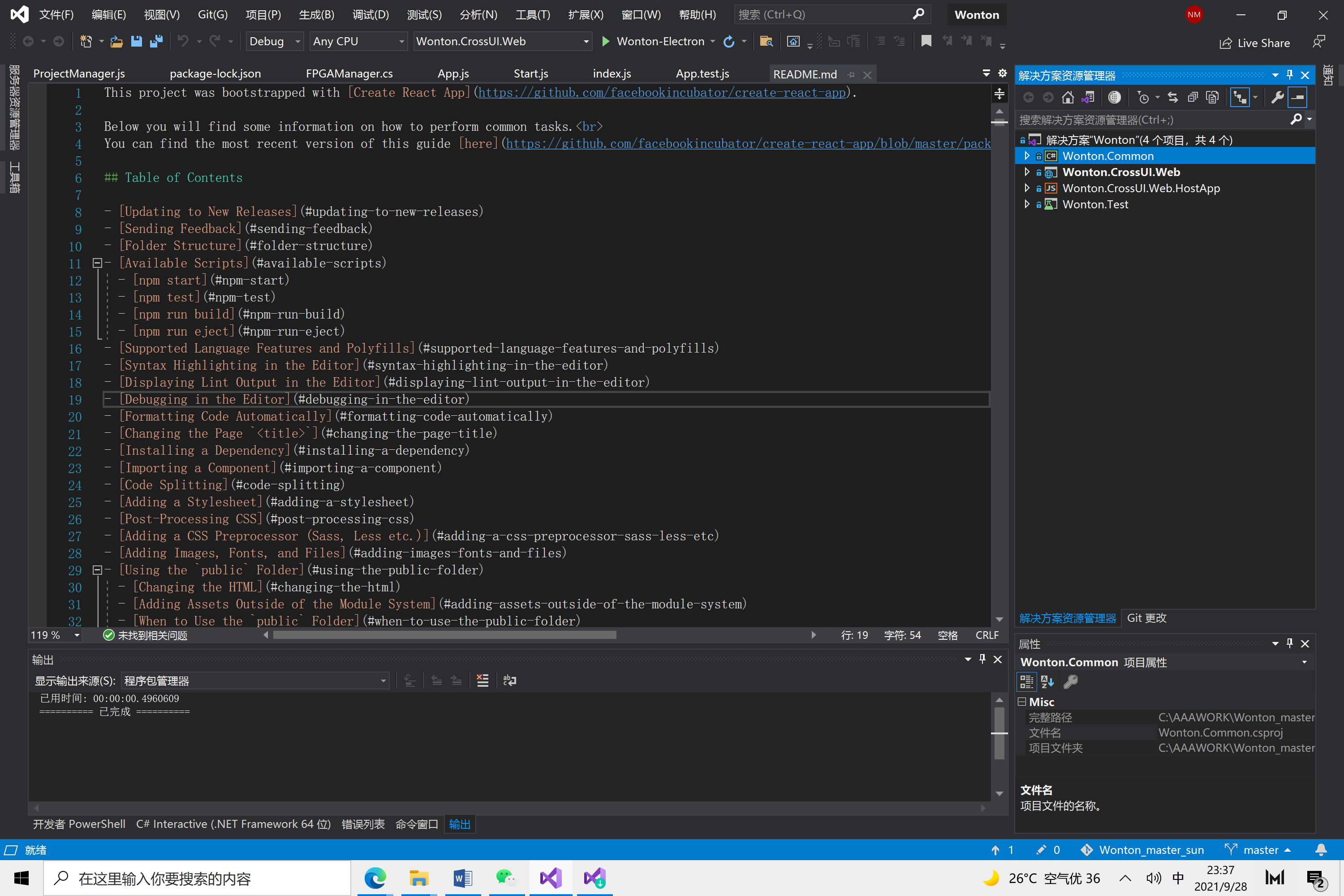
安装时记得装上以下勾上的框架：





## 2.2开始以调试模式运行wonton

安装好visual studio后，到github上下载wonton的源码。找到wonton.sln文件，点击以visual studio打开，你应该能看到以下页面。



下一步，为了在你的电脑上运行一个nodejs的程序，请在你电脑上下载好nodejs框架，以及其包管理工具npm。这是我电脑上的版本号，理论上大版本一致就不会有问题：



请注意，你现在依旧不能成功运行wonton。因为你下载下来的只是源码，需要编译过一次后才可以进行运行。

Win环境下开一个cmd窗口运行build.cmd即可进行编译，更推荐你打开powershell运行build.ps1，这个ps1实际也是调用build.cmd，不过显示格式更友好。

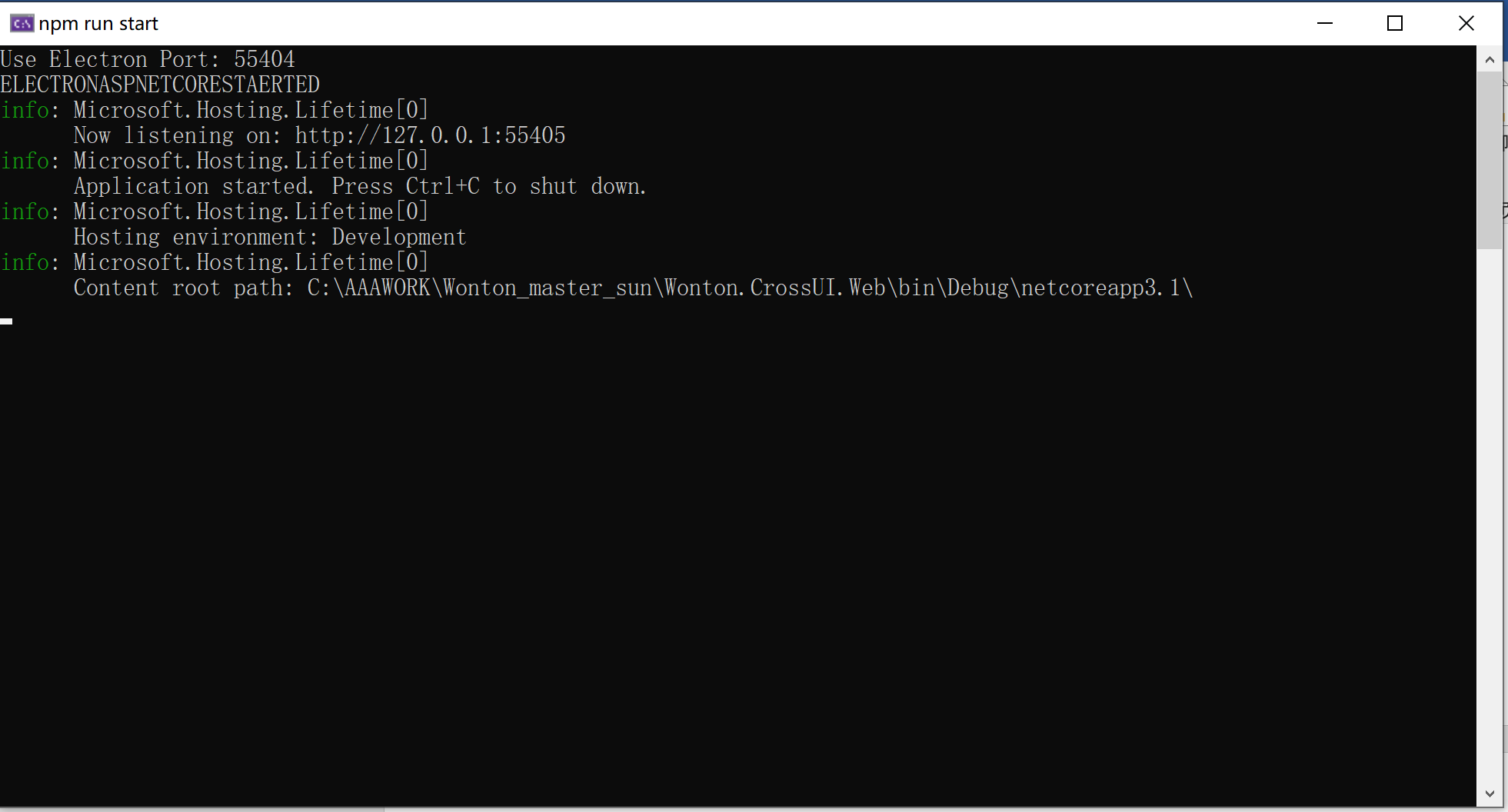
当然，实际上这一步就很可能出各种环境问题，所以我干脆上传了我编译成功过后的wonton文件夹环境：

<https://pan.baidu.com/s/1rMAPhLk-nscLvQAh8-oMfQ> 提取码：wton

现在，将visual studio配置如下：

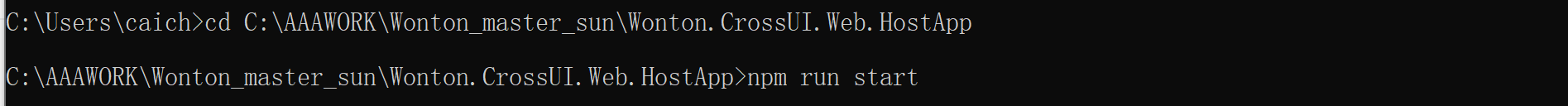


点击运行按钮，应当弹出以下界面：

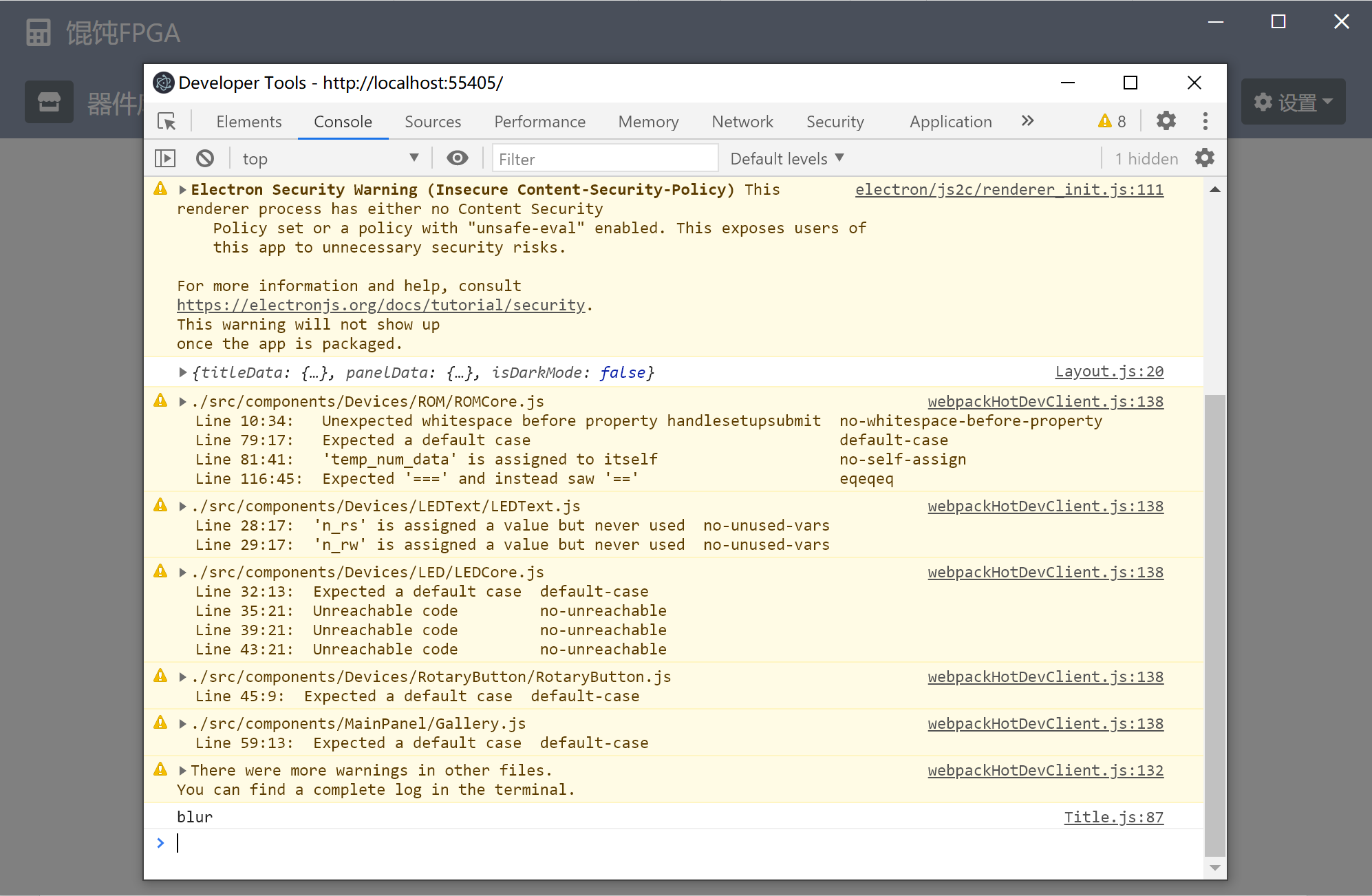


这说明后端已经开始工作了。

现在，新开一个cmd窗口，切换到Wonton.CrossUI.Web.HostApp目录下，输入npm run start运行：



这时应当正常启动wonton软件：



现在，你可以像正常使用wonton一样对软件进行调试。

开发者工具会自动弹出来，如果你以前有在任意一个浏览器中右键点开检查，你会发现开发者工具其实就是浏览器自带的页面检查工具，这是因为wonton前端本身就是一个类web的框架。

你可以试着修改wonton中的js代码。

**添加器件所需要修改的所有代码均在visual studio右边资源管理器页面的Wonton.CrossUI.Web/ClientApp/src/components目录下。**

你可以做一个小尝试：点开title.js的475行，将 馄饨FPGA 修改成任意字段，切换回wonton界面，在几秒后你应当可以看到软件左上角的 馄饨FPGA被替换成了你修改的字段。没错，javascript是实时渲染的，你不需要重新运行调试就可以实时看到你修改代码造成的影响。

当然，请注意在修改代码前停掉当前verilog的运行，这是因为后端并不能实时渲染。少部分修改代码造成的错误也可能造成在本次运行中出现不可逆的错误，碰到这种情况请重新运行项目的前后端。

# 三、添加器件

## 3.1器件简介

### 3.1.1 Input器件

器件源码在devices目录下，每个器件由三个部分组成，以输入器件按钮为例：



ButtonCore是器件的核心逻辑，Button将Core进行封装，并规范接口与FPGAManager交互，Css主要调整器件中每个单元的样式。

**ButtonCore源码如下：**



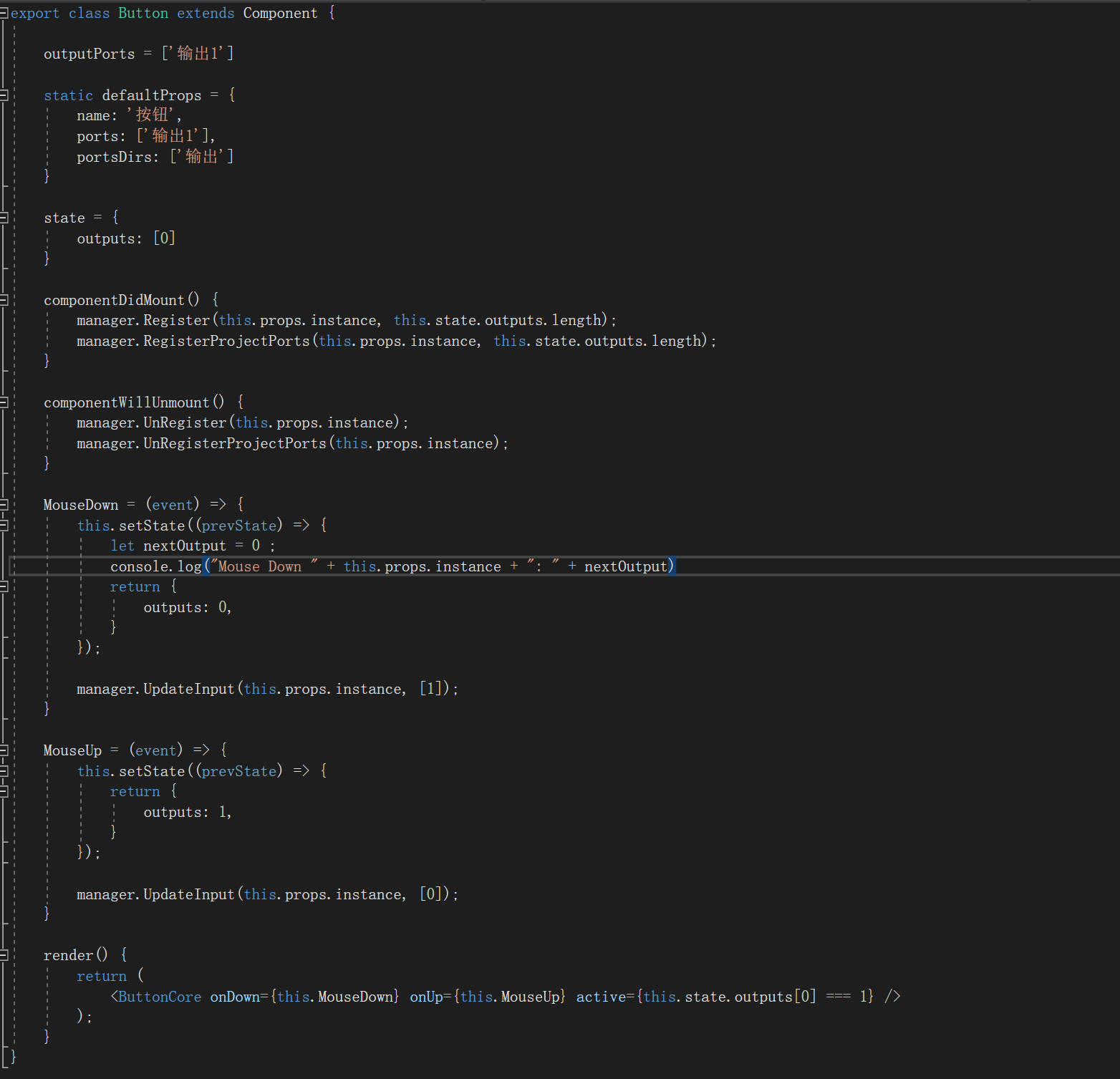
属性defaultProps中：

name和ClassName是你为器件定义的名字，size中四个元素分别是：默认宽，高和最小宽高，控制器件布局到layout上的尺寸。

onCheckMouseDown和onCheckMouseUp：

定义用于处理鼠标按下和放开的响应，其具体功能通过Button.js中的回调函数实现。

**Button源码如下：**

****

defaultProps中：

name是器件名。ports和是端口名称，是一个数组类型。portsDirs与ports一一对应，是端口类型。（outputPorts为FPGAManager中将硬件端口与软件端口对应所需要的类，只在INPUT器件中需要声明）

componentDidMount/componentWillUnmout：

器件被创建和移除时会调用的函数，器件中调用了FPGAManager类中的函数将器件安装或移除到端口上。

MouseDown/MouseUp：

实现功能的回调函数。

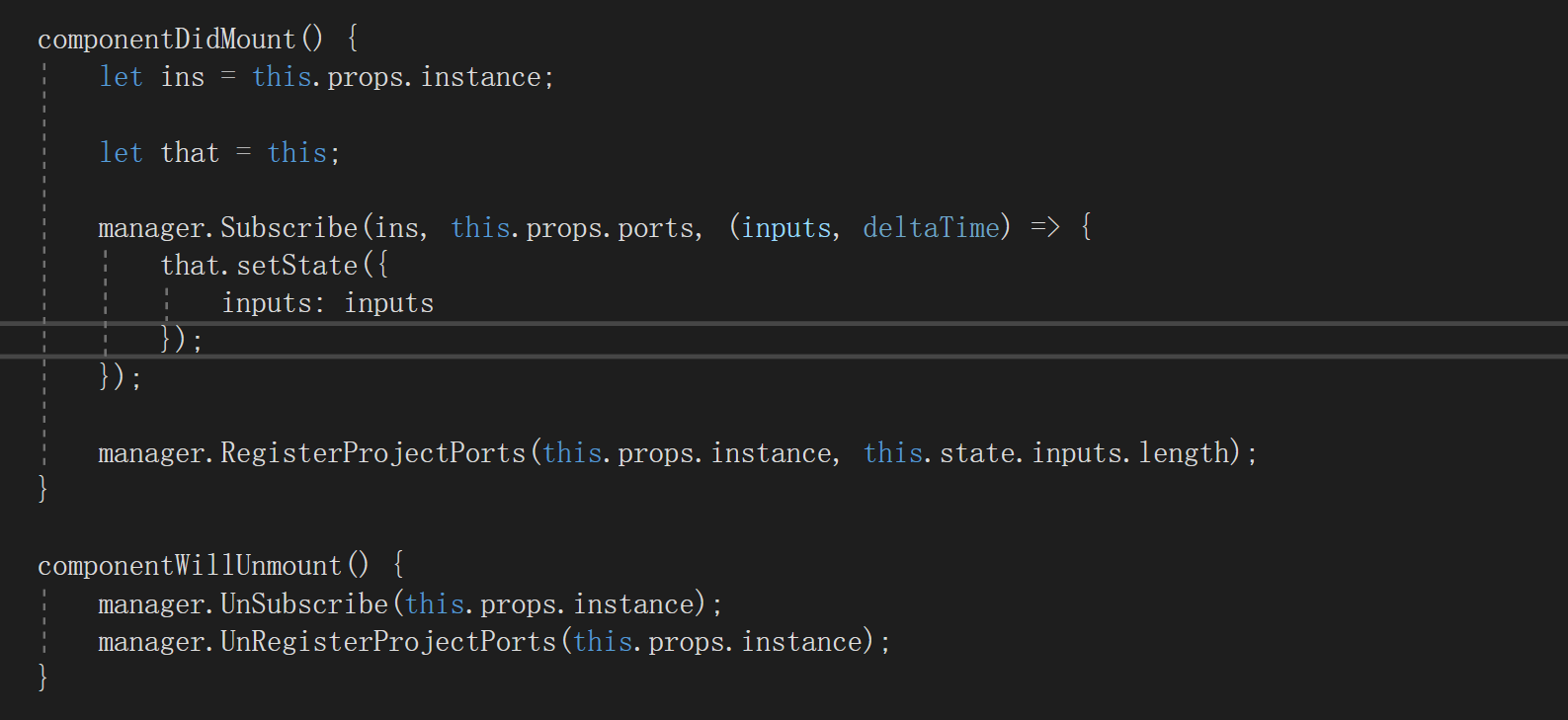
manager.UpdateInput：

更新输出（软件的输出，FPGA的输入），第一个参数指向被创建的本instance，第二个参数是数组类型，要更新的传入值，位置与outputPorts一一对应。

### 3.1.2 output器件

Output器件与Input器件大体相同。

唯一的不同点在于，以LED器件为例：



Subscribe/UnSubscribe：

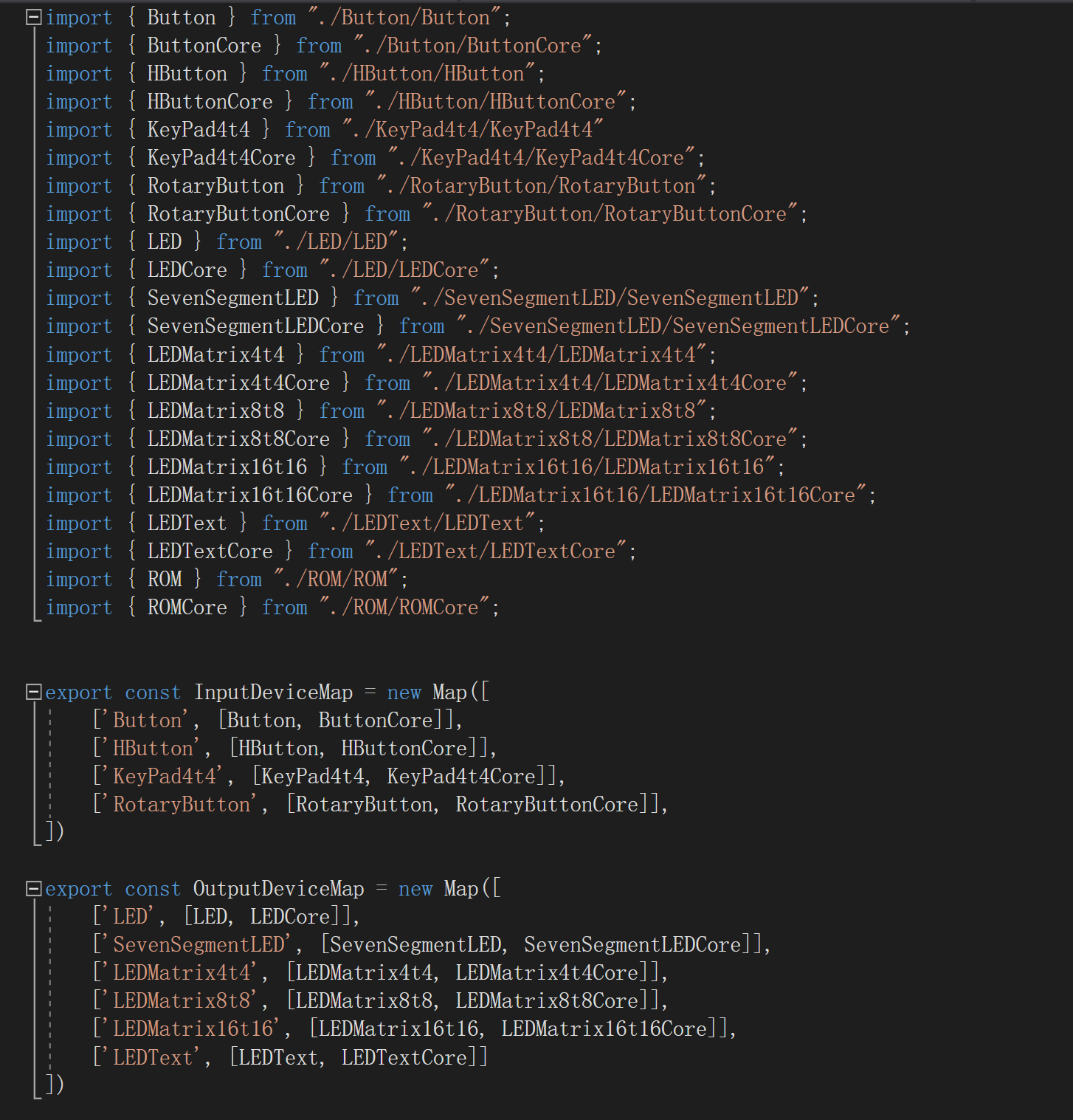
器件通过FPGAManager开始/结束对FPGA的端口进行监听。在Subscribe时，检测到端口传入的数据变化后，会执行其中的函数。

## 3.2 如何添加你自定义的器件

### 3.2.1 创建器件代码

照着其他器件的源码写框架，加上你自己写的javascript逻辑和前端页面。

### 3.2.2在Devices.js中声明



## 3.3 为你更新过后的wonton制作安装包

运行build.cmd或build.ps1即可。加上-useMagic会使用国内网中的资源，不加默认从官网下。

你可以打开build.cmd（win下的自动打包脚本）或build.bash（linux/Mac下的自动打包脚本）查看其中的内容。实际上脚本做的无非是帮cake（一个打包程序）配环境，运行时会多出tools文件夹，下面放了编译需要的依赖，而cake本身相当于一个黑盒。之后cake会检测当前的平台自动打包生成对应平台的安装包以及可执行程序。生成的新安装包和可执行程序均放在build文件夹下。

如果只是对源码进行小改动，直接在2.2中上传的文件夹下编译即可。

如果编译失败，大概率是build.cmd/build.bash下的依赖没有下好。

检查tools目录下是否有空文件夹。

检查build脚本中使用的网址是否可用。

检查终端以及build脚本中的下载相关命令行是否可以连接到网址。

检查build脚本中的一些命令是否正常工作，bash特别注意curl -s使用的是静默模式，下载失败不会提醒。

如果安装的某个步骤中提示node\_modules目录下缺失文件，则是依赖没有装好。

package.json中会说明所需的dependency及其版本。

最坏的情况是一直跑不出来，那就在调试环境下跑Verilog也是可以的，需要注意的是这个模式下wonton和FPGA的交互会有一定的延时。

# 四、其他

wonton软件离不开老师和几位师兄的帮助，在此郑重致谢。

如有使用问题或bug请到github上提交，[联系wontonfudan@163.com](mailto:联系wontonfudan@163.com)或本人微信。