基于OpenWrt的智能家居控制中心的设计与实现说明文档

1. 智能家居控制中心简要分析

智能家居控制系统实现的平台分别有路由器和云服务。

用户可以通过手机浏览器凭用户名密码进入系统，对受控设备进行控制。路由器这端调用系统管理模块提供的API服务，获取数据后对数据进行解析处理，发送控制指令给受控设备从而完成相应的操作。

1. 硬件选择

路由器开发板 mt7620a

五路蓝牙继电器开关

USB蓝牙适配器

USB转串口线

网线

5v电源

线材若干

1. 路由器开发环境搭建

1，编译OpenWrt固件

首先配置好编译的环境。参考OpenWrt官方网站说明。安装依赖（官方推荐是Debian/Ubuntu）

命令如下：

sudo apt-get install subversion g++ zlib1g-dev build-essential git python rsync man-db libncurses5-dev gawk gettext unzip file libssl-dev

Ubuntu对于新手容易上手，所以推荐使用Ubuntu。可以采用虚拟机安装Ubuntu或者笔记本安装双系统，看个人情况而定。

这里我实际操作中使用的是VPS上的Centos7编译OpenWrt。云服务编译比较快，一般在本机或者虚拟机编译OpenWrt需要很长时间的等待。

依旧要安装依赖。

命令如下：

yum install subversion binutils bzip2 gcc gcc-c++ gawk gettext flex ncurses-devel zlib-devel make patch unzip perl-ExtUtils-MakeMaker  glibc glibc-devel glibc-static quilt ncurses-lib sed sdcc intltool sharutils bison wget git-core openssl-devel xz  **zlib-static**

接下来便是拉取OpenWrt源码。下面的操作必须要在普通用户权限下进行。

命令如下：

git clone <https://github.com/hi-wooya/openwrt-hiwooya-stable.git>

下载完后进入该目录。

命令如下：

cd openwrt-hiwooya-stable

接下来输入以下命令：

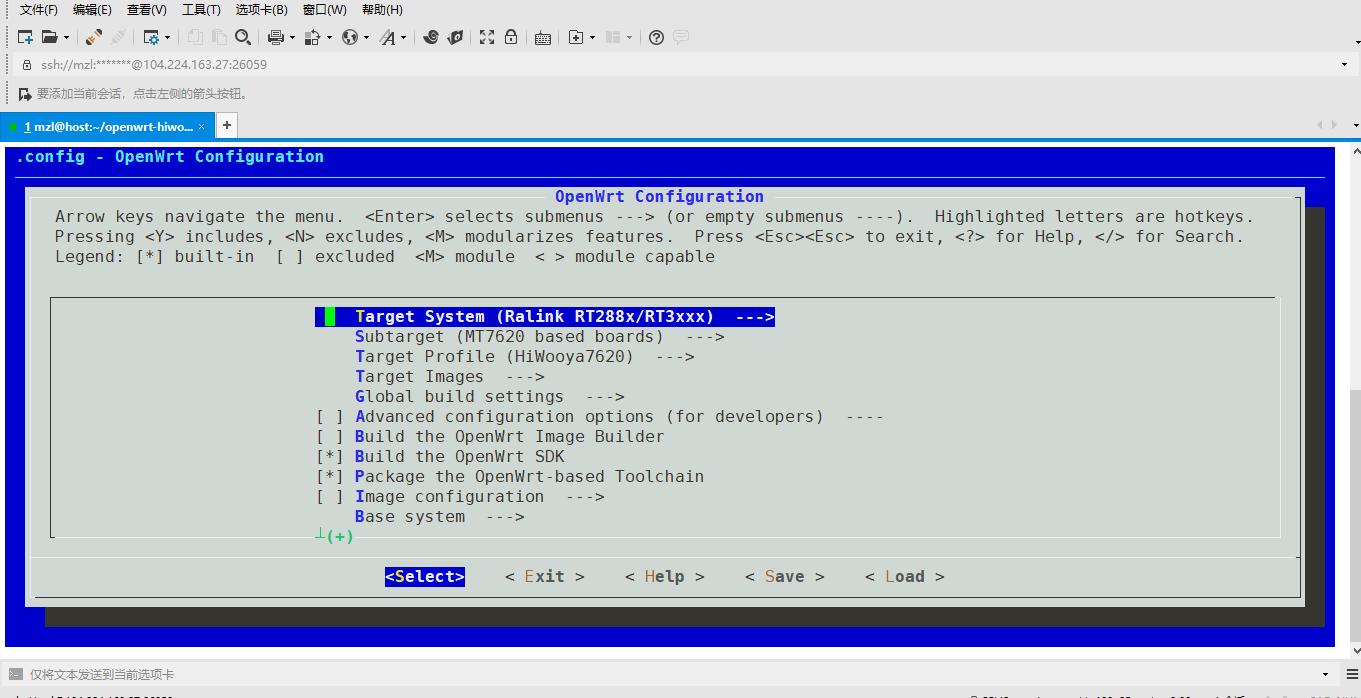
./scripts/feeds update -a

./scripts/feeds install -a

make defconig

make menuconfig

前面两条命令安装feeds包，下载完成就好，第三条命令检查编译环境的依赖，没有问题，就可以进行第四条命令进行固件的编译选项配置。



这里选择：

“Target System” ⇒ “Ralink RT288x/RT3xxx”

“Subtarget” ⇒ “MT7620 based boards”

“Target Profile” ⇒ “HiWooya7620”

接下来就是输入命令编译了：make

由于是远程连接vps的，所以，编译时间过长，一旦断开连接，会导致编译不成功，所以让编译命令始终在后台运行。

修改命令为：

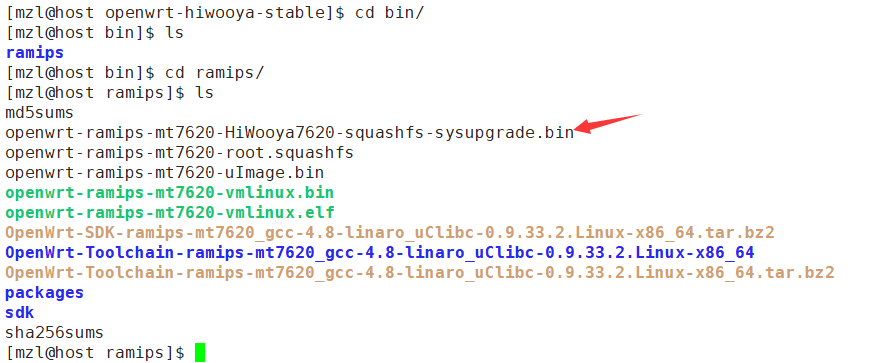
nohup make -j1 V=s &

在当前目录下会生成nohup.out文件，这里会记录编译的详细情况，出错的话也能知道哪里出问题，从而修改。

编译一般需要几个小时的时间，取决机器本身的配置和网络状况。

编译完成之后，会在bin目录下生成一些目录和文件。

建立github仓库，将编译好的固件上传到仓库，方便将文件下载到路由器开发板上。



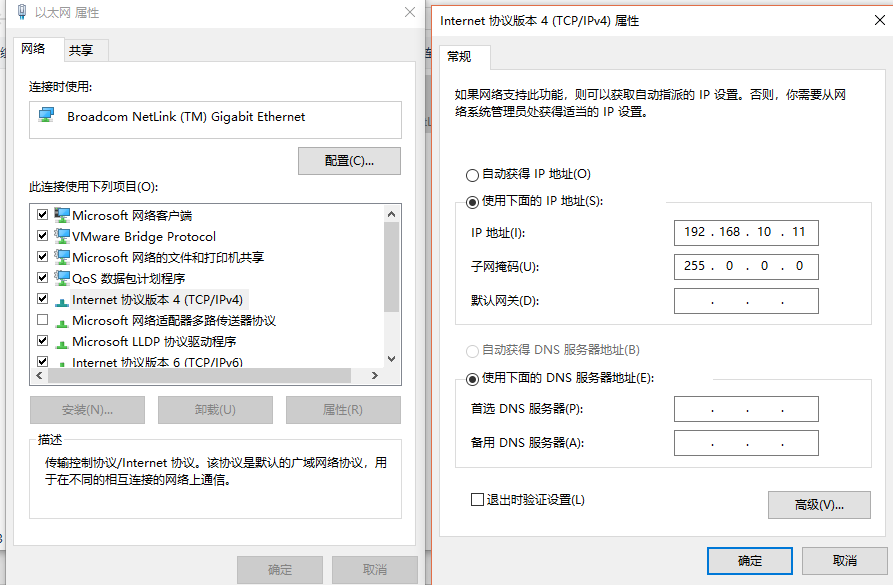
2，路由器开发板刷写固件并配置路由器网络

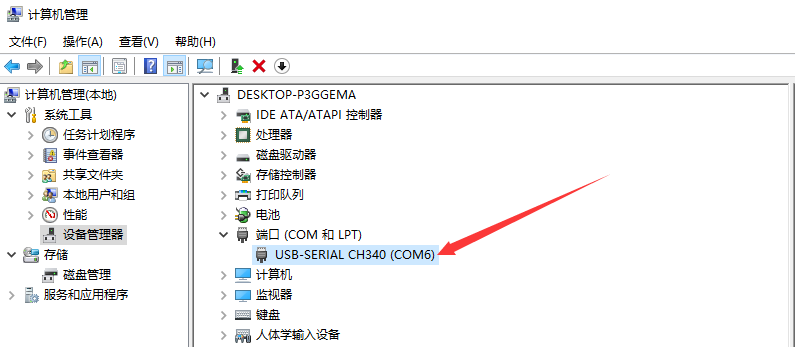
配置Windows网络静态ip地址。

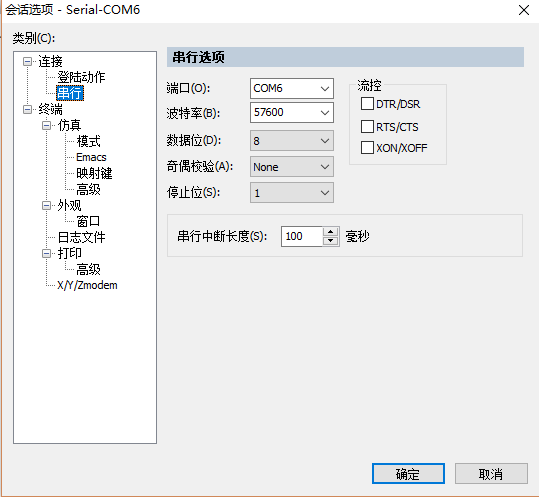
在Windows系统下，下载git仓库上的固件，运行SecureCRT。

系统没有串口驱动的话，需要安装串口驱动。可以用驱动精灵进行安装。网址是<http://www.drivergenius.com/>。

查看并配置串口。



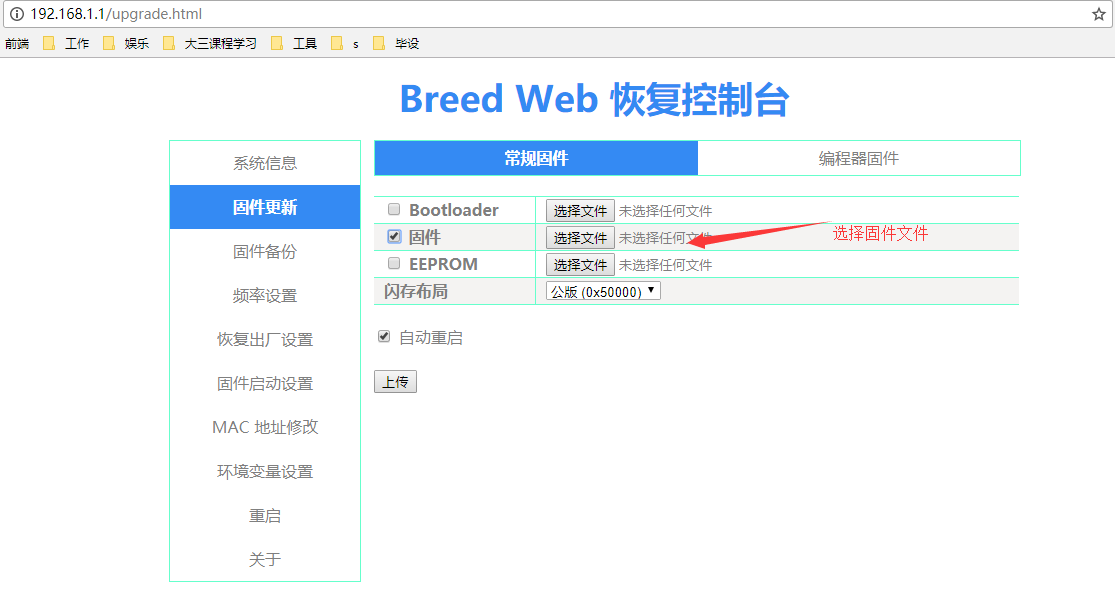




连接后按任意键，进入breed模式，浏览器输入192.168.1.1，进入breed web恢复控制台网页。



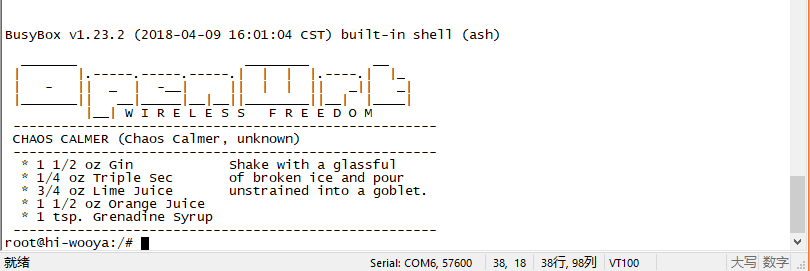
选择下载好的固件进行升级系统。



注意：上传固件文件会很慢，所以需要耐心等待。

升级固件完成后，重启。

这次通过启动SecureCRT，开发板供电，不要再按任意键，默认进入系统。等待一小会儿之后，回车就可以看到OpenWrt的欢迎界面。



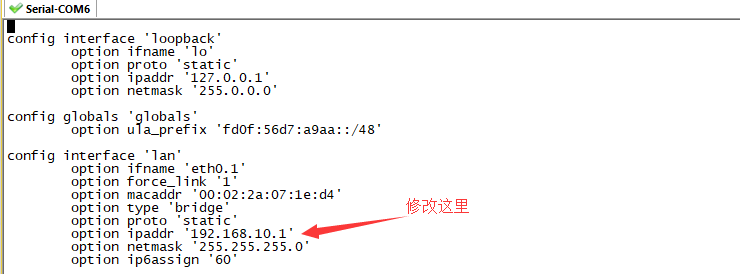
设置开发板IP地址。

控制台输入ifconfig命令来查看开发板的IP地址，可以看到开发板的默认IP地址。

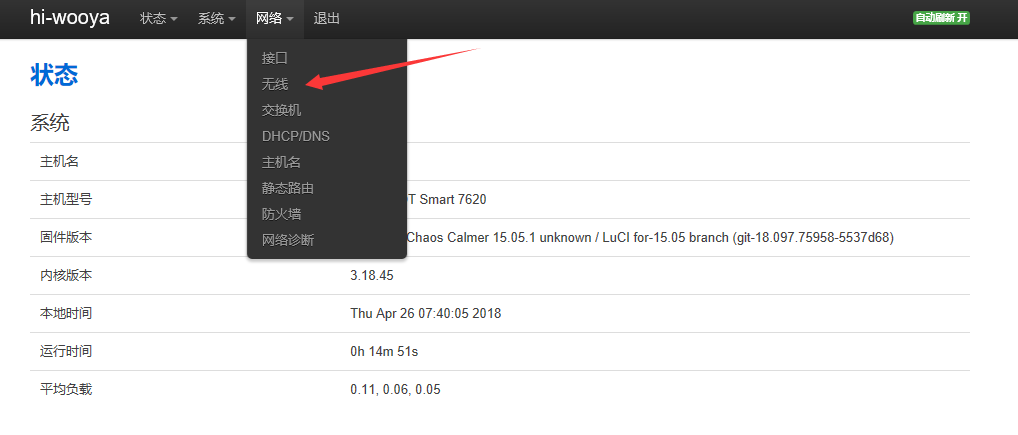
修改IP地址，修改/etc/config目录下的network配置文件。

命令如下：vim /etc/config/network

修改完成后，保存退出，重启开发板，就会发现IP地址已经修改。这里修改为192.168.10.1。



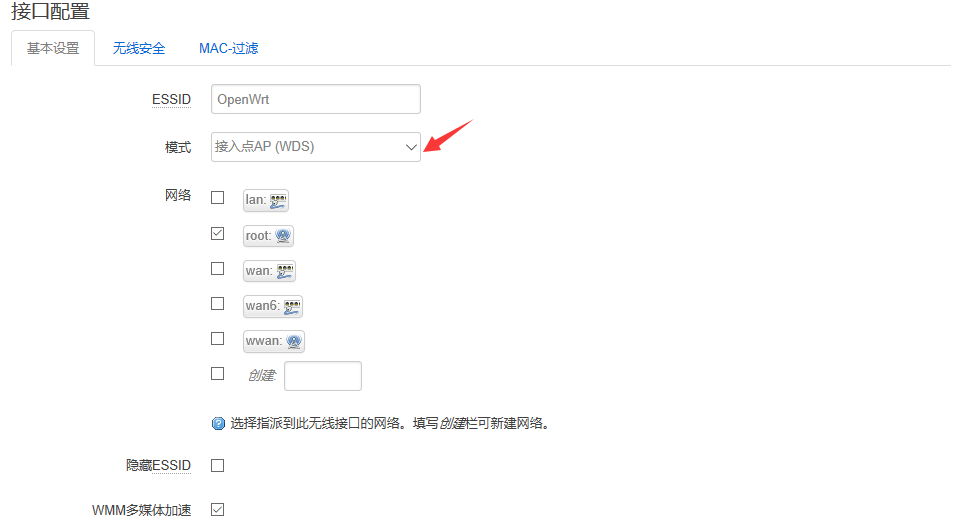
修改完成，重启开发板，PC上就可以在浏览器地址栏输入192.168.10.1这个地址，进入openwrt系统，初始时是没有密码的，之后自己设定密码。现在的开发板发射的WIFI是没有网络的，是不满足开发需要的。所以这里需要配置网络。

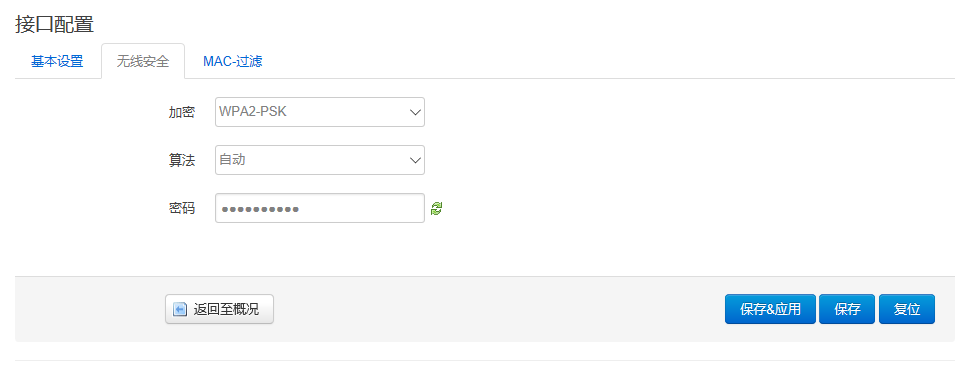




进入无线这里配置，首先是搜索网络，然后加入，这里输入所要加入的网络的密码即可。

选择接入点AP(WDS)，无线安全这里设置连接该AP的密码。



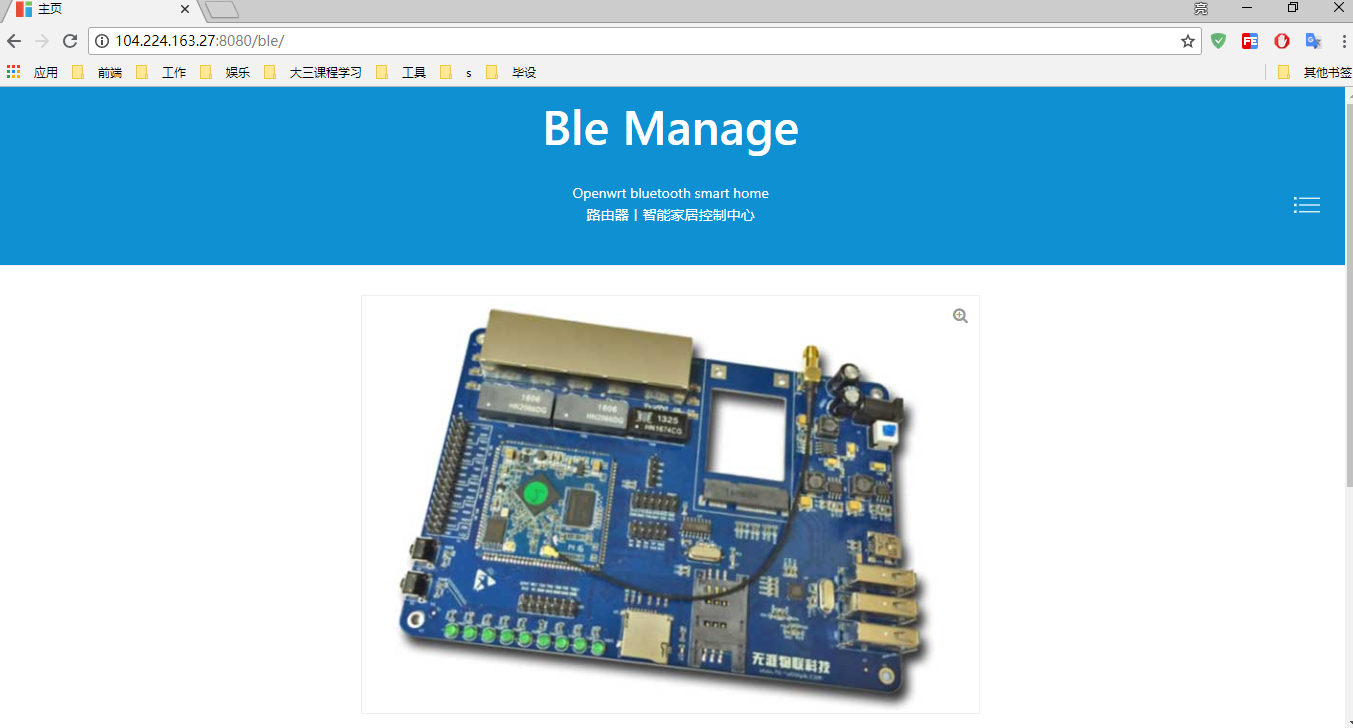


尝试手机连接该AP,是否可以上网，如果可以，说明配置成功。也可以在OpenWrt系统中利用ping命令测试网络是否通畅。

检测网络正常后，就可以进行后面的开发了。

1. 外网控制中心系统管理模块

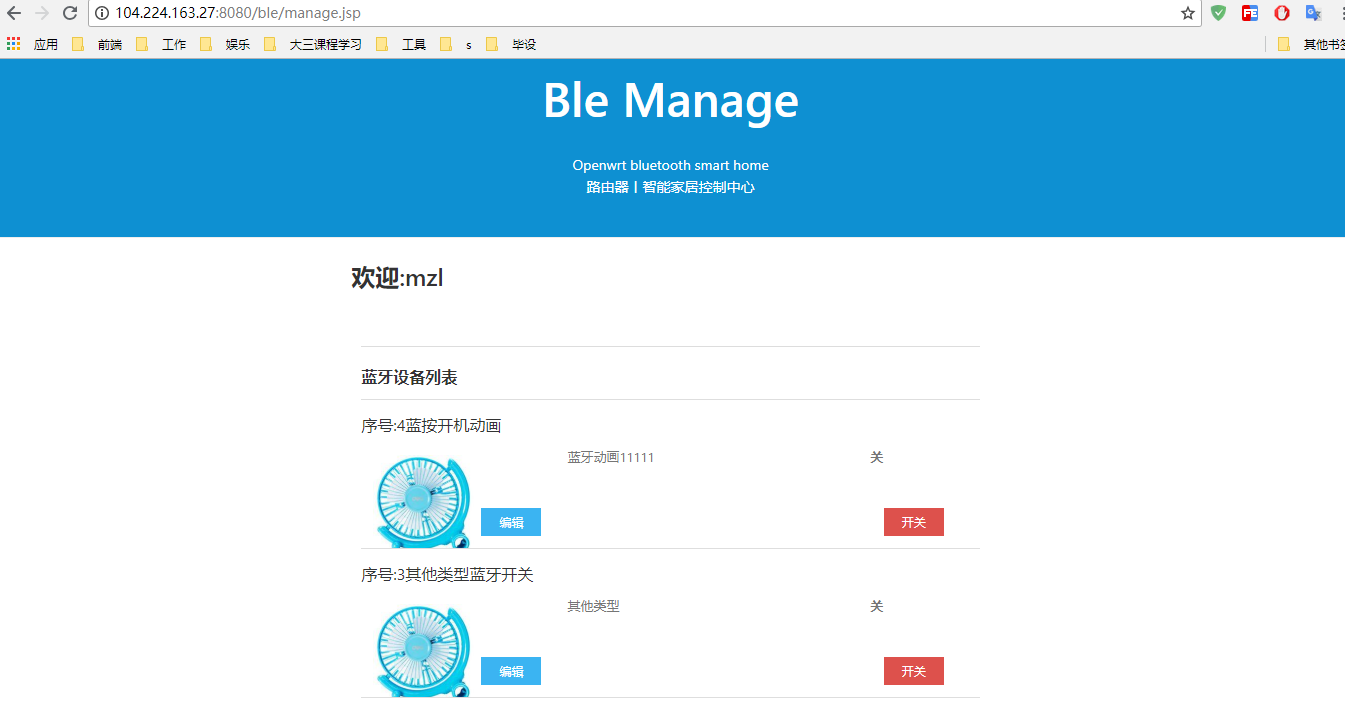
1，系统界面



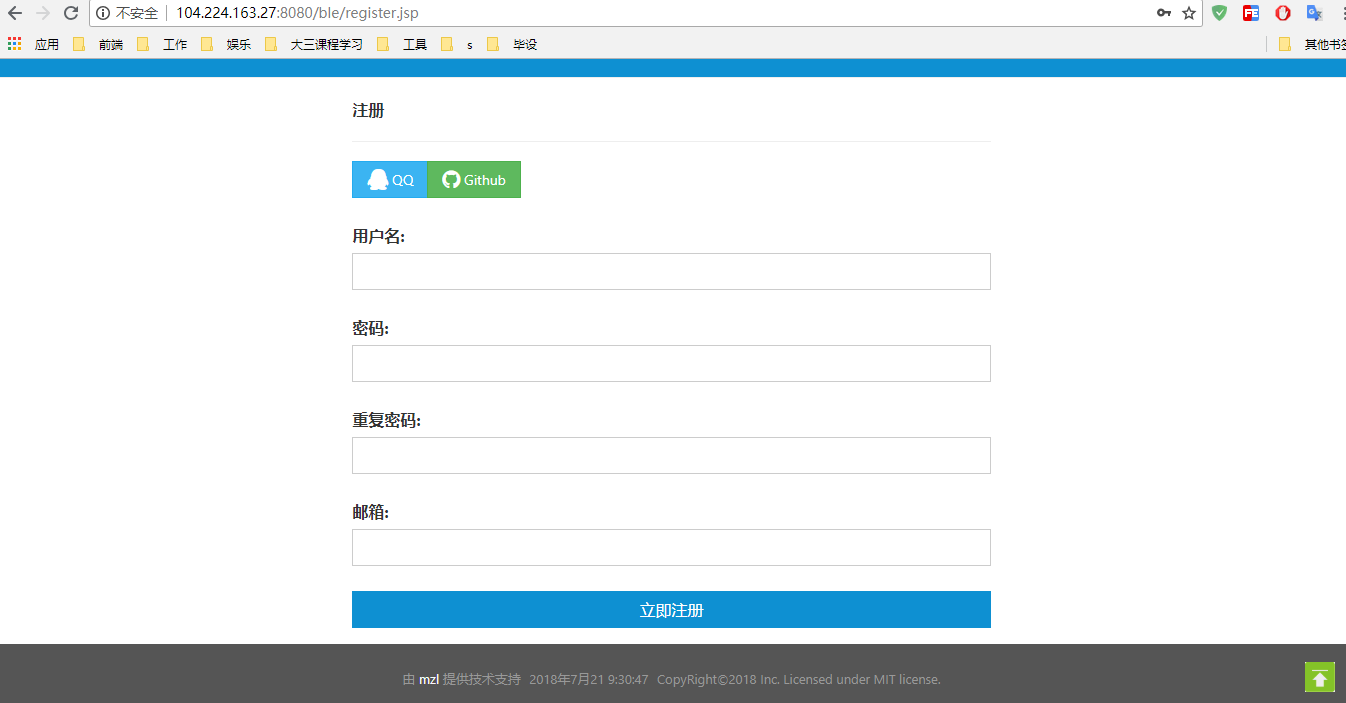
个人开发的外网控制中心系统部署在云服务上，项目源码文件在附件中，名称为ble文件夹，可以在导入eclipse中进行后续开发。

这里给出一组用户名密码，之前自己做测试用的。

用户名：mzl 密码：123456



也可以重新注册一个账号进行测试。



2，系统API

蓝牙

http://104.224.163.27:8080/ble/servlet/QueryDevicesServlet

数据返回格式为json。

ZigBee http://104.224.163.27:8080/ble/servlet/QueryZigbeeByLately?terminal\_id=0001

数据返回格式为json。

3，系统部署

这里建议使用我部署的服务。当然也可以修改源码之后重新选择一个服务进行部署，之后路由器这块软件包的编写需要修改，重新编译IPK，重新在OpenWrt安装，这样也是可以的。

1. 路由器开发板软件包制作和安装

以蓝牙控制为例进行说明，其他类型设备在路由器开发板上开发流程类似，主要修改核心功能，基础的模式和蓝牙控制的一致。

首先补充一点，关于路由器蓝牙适配器和蓝牙受控设备的匹配问题，解决如下：

这里选用Ubuntu系统，笔记本安装了双系统。

5路蓝牙继电器开关接入电源。启动Ubuntu系统，笔记本usb插入蓝牙适配器。进入蓝牙设置。

点击5路蓝牙继电器开关的蓝牙名称，进行配对。输入PIN码，一般是0000或者1234。

在/var/lib/Bluetooth/xx:xx:xx:xx:xx:xx/xx:xx:xx:xx:xx:xx目录下面会生成一个info文件，这个文件中包含了蓝牙匹配的KEY的相关信息。需要将其备份到git仓库，并查看其文件和目录的权限。

后面路由器蓝牙适配器和蓝牙受控设备之间进行匹配需要这个文件。还有就是需要一个脚本文件配合软件包完成匹配。这里说明一下hci0就是路由器开发板所插入的这个蓝牙适配器设备。

ble.sh:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | cp -r 20:16:06:15:06:15 /var/lib/bluetooth/00:1A:7D:DA:71:13 | |  |  |  | | --- | | hciconfig hci0 up | |  |  |  | | --- | | hciconfig hci0 lm master | |  |  |  | | --- | |  | |  |  |  | | --- | | mknod /dev/rfcomm0 c 216 1 | |  |  |  | | --- | |  | |  |  |  | | --- | | chmod 666 /dev/rfcomm0 | |  |  |  | | --- | |  | |  |  |  | | --- | | rfcomm bind /dev/rfcomm0 20:16:06:15:06:15 | |  |  |  | | --- | |  | |  |  |  | | --- | | /etc/init.d/bluetoothd stop | |  |   /etc/init.d/bluetoothd start |

注意：有个小问题是，在OpenWrt系统开启后，脚本执行不成功，需要开启蓝牙适配器，也就是上面的第二条命令：hciconcig chi0 up。让它的状态为开启状态。

接着在编译OpenWrt固件的环境中进行编写蓝牙软件包，其他类型的没有之前的一个匹配问题。以蓝牙模块为例：

在openwrt-hiwooya-stable/package/utils/目录下创建新的目录，这里是ble目录。结果如下：

openwrt-hiwooya-stable/package/utils/ble

具体的源码文件在附录中，打开后在src目录下修改核心的功能文件。这里只能用C语言进行编写。编写完成后修改对应的Makefile文件，进行make 编译出ipk软件包。

具体的命令运行要返回openwrt-hiwooya-stable这一级目录，命令如下：

make package/utils/ble/compile -j1 V=s

没有报错的话，会在当前目录下的bin/ramips/packages/base/中生成一个ipk文件。这个就是我们要安装到路由器开发板上的文件。运行Opkg install 软件包名称，完成安装。如果编译软件包出错的话，需要修改源文件或者Makefile，直到不报错为止。

如果安装的ipk软件包并不能满足开发的功能，Opkg remove 软件包名称 ，卸载该软件包。重新上述操作，修改源码，生成ipk文件。接着再安装，测试。

测试的环节，在路由器开发板OpenWrt系统终端中输入该软件包的名称命令，即开启该软件。比如蓝牙，这个命令是ble。

整个的开发流程就是这样。

1. 注意事项

摄像头挂载资料链接：

<https://segmentfault.com/a/1190000011802017>

这里要做的话请参考上面的链接内容，路由器将在本地网络下下载这个拍摄的图pain保存在本地，然后上传到外网云端的MySql数据库上，外网管理模块读取数据库中摄像头拍摄的图片数据，在云端项目目录下创建图片文件后，前端访问得到展示图片的内容。

还有一点是最开始进入OpenWrt系统时摄像头功能模块时没有开启的，所以需要进入上面的链接了解后进行操作。

比如运行这条命令开启：

mjpg\_streamer -i "input\_uvc.so -f 15 -r 320\*240 -d /dev/video0" -o "output\_http.so -p 8080 -w /www/webcam/"

在进行ZigBee开发调试的时候，首先要进行协调器联网的操作，这个时候短路帽的接法是一种，而正常的ZigBee设备运行是则是另外一种接法，这个请看百度网盘中提供的ZigBee资料视频。还有就是ZigBee设备的固件目前是不用在刷写的。使用方面的需要这一点。短路帽的接法，否则不能切换，导致路由器和协调器之间无法通讯。

1. 附件

个人OpenWrt开发的GitHub仓库地址：

https://github.com/mzl123/openwrt

git clone https://github.com/mzl123/openwrt.git

百度网盘资源：

链接: https://pan.baidu.com/s/1sjV7663nwiVILDc0\_O5NUA 密码: mavk

资源自取。

注意：ZigBee参考资料请下载那个大的压缩包文件，下载后请查看最终篇部分的内容，具体的操作请参考这个里面的资料。

1. 操作

前面主要讲的是一个开发流程，鉴于现有程序，接下来介绍一下如何将这些设备进行组装。上电开启设备就不说如何操作了。

设置PC机静态IP地址。上面在刷写路由器开发板OpenWrt系统已经提到，按照上面的步骤进行操作。配置无线网络，使路由器开发板网络通畅，也是进入Web界面进行配置。

Xshell工具进入路由器开发板OpenWrt系统。下载和安装IPK软件包，可以到GitHub仓库上下载，也可以自己重新编译。鉴于有一个板子是已经安装好这些软件的。密码是123456或者root，如果没有记错的话。如果错误的话，我没有遇到这种情况，所以也不知道该怎么解决。比较笨的方式就是重新选择刷写固件，重新配置，重新安装软件。IPK文件在附件中有。那些是最后一次更新的软件包。

个人编译的固件放在GiHub上，地址：

<https://github.com/mzl123/openwrt/blob/master/bin-dir/openwrt-ramips-mt7620-HiWooya7620-squashfs-sysupgrade.bin>。

安装完成后就是运行命令进行调试，比如蓝牙这里命令就是：Ble。执行完毕就准备调试。在外网管理系统中进行操作，受控的蓝牙设备被控制，说明成功完成这个操作。会有一定的延时，请耐心等待。而摄像头模块只能查看，外网管理系统没有明确写明，需要访问请输入一下地址：<http://104.224.163.27:8080/ble/webcam.jsp>。最后的ZIgBee模块功能暂时未开发完成，用的传感器是温湿度传感器，是蓝色的那个。具体是哪一个请参考资料进行选择。完成的是读取一个终端的温湿度数据。还是需要上面提及一下上面的注意事项。协调器进行联网配置和正常的ZigBee模式，短路帽的接法还是不一样的。具体细节请看视频资料。这里由于时间有点久了，也没有实物，说不清楚。

到这里整个说明文档就结束了。如果有什么问题的话，请加QQ:1059115590。或者发邮件到QQ邮箱。将问题状况描述清楚或者加上截图，看到后会及时给出解决的参考意见