

网页控制车库电动门开关以及车库灯

设计目标

大多数车库门内部都由一个按键开关控制，基本上是按一下开启，再按一下关闭，该控制器通过网页操作一个和该开关并联的继电器，每点击网页上的车库门按钮一次，该继电器将闭合 0.5 秒，实现通过网页开关车库门的目的，为了增强安全性，该按钮加上了密码保护。车库灯的功能也和车库门功能相关，如果车库灯处于关闭状态，每次开关门会自动开启车库灯 3 分钟，方便车人进出，网页上的车库灯开关按钮具有优先控制权，并且会取消因为车库门开关触发的 3 分钟亮灯功能。

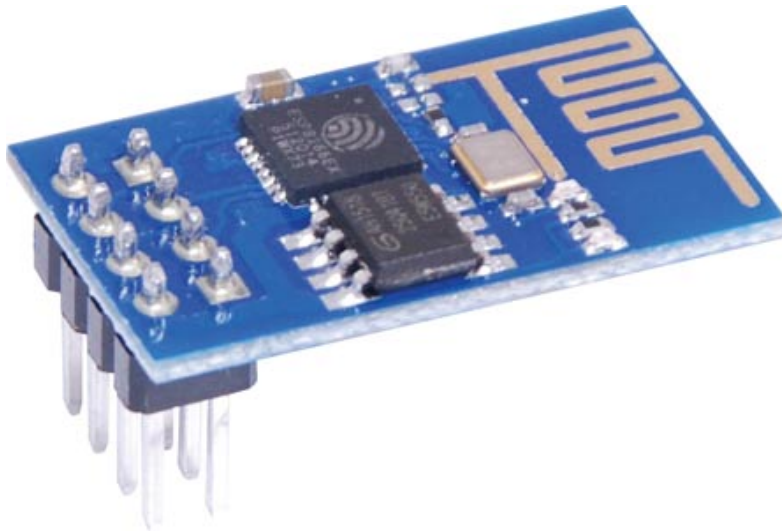
该控制器由 ESP8266 NodeMCU 通过 LUA 代码来实现 WiFi 连接，网页传送，POST/GET 数据处理，以及相关逻辑时序控制。



控制网页图

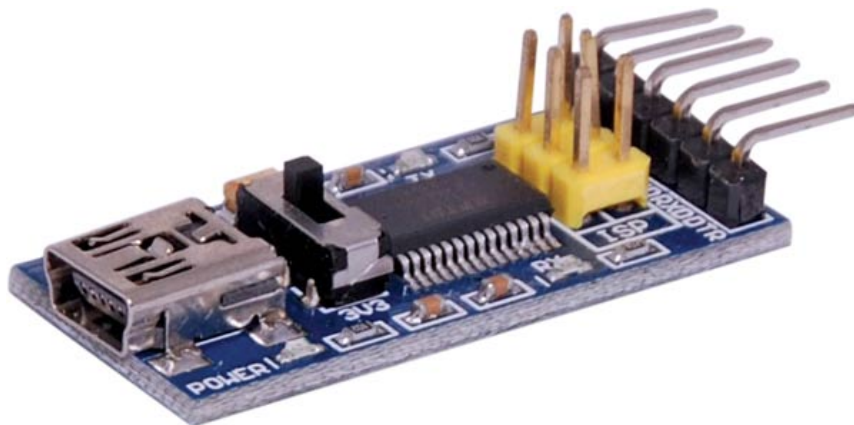
硬件设计

ESP8266 无线 IoT 模块



这是设计的主要部件，负责处理所有 WiFi，TCP，HTTP 和控制脉冲信号。可以从许多渠道买到，比如淘宝或者 ebay（查询 ESP8266），以及当地的一些电子器件零售商店，比如 www.altronics.com.au。

FT232RL USB 串口转化器

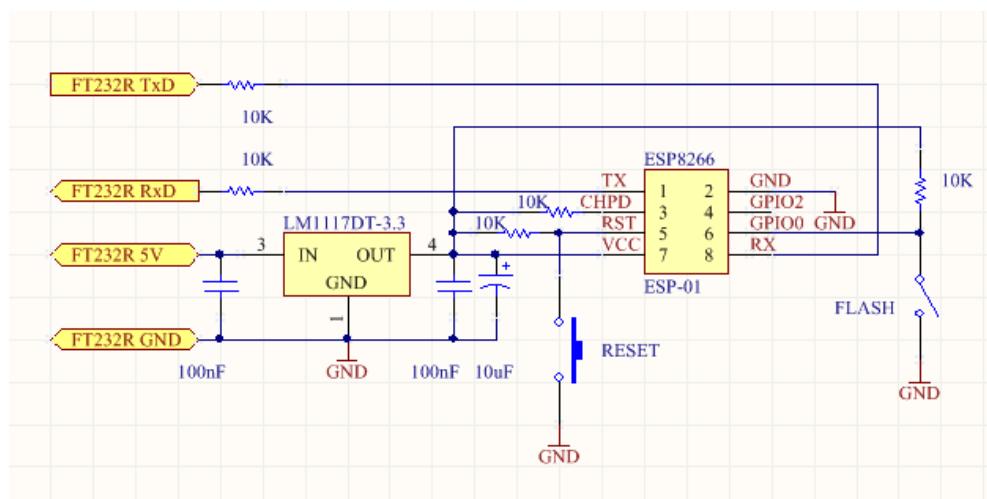


这是用来将控制程序写入 ESP8266 的器件，和 ESP8266 一样，可以从淘宝或者 ebay（查询 FT232R），以及当地的一些电子器件零售商店，比如 www.altronics.com.au 买到，转换芯片技术资料：

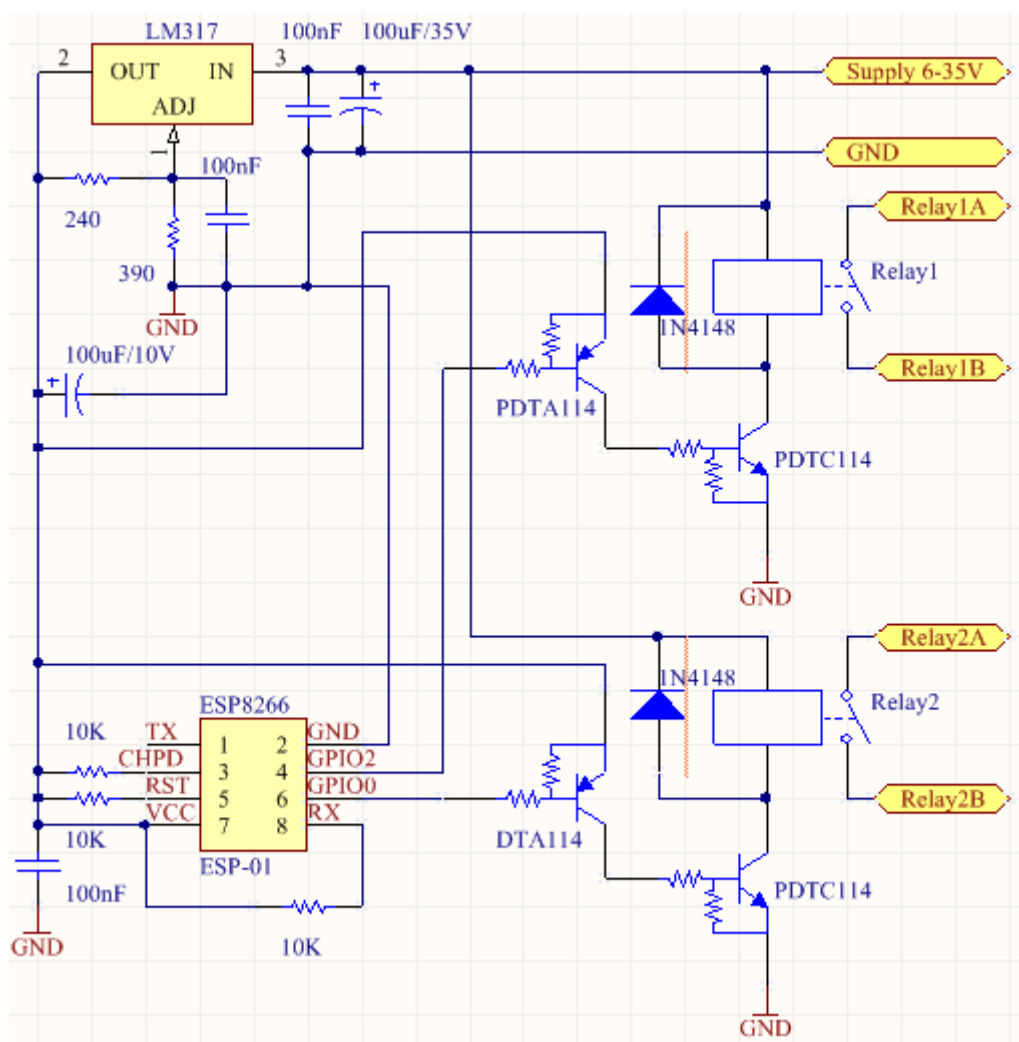
http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232R.pdf

其他零件以及辅助电路

请参考编程器连接电路图和车库电动门控制电路图。



FT232RL 和 ESP8266 连接图



车库控制器原理图

车库控制器原理图中继电器 Relay1 和 Relay2 需要根据供电电压 Supply 选择。LM317 比 LM1117 更能适应宽输入电压范围，可以装在更多的车库门电动控制器上，许多车库门电动控制器内部都使用 24V 直流电。

软件设计

测试编程器连接

按 FT232RL 和 ESP8266 连接图连线后，需要测试 PC 和 ESP8266 的通讯，减少硬件连接错误造成的麻烦。

第一步是通过 USB 电缆连接 FT232RL 和 PC 并安装 FT232RL 驱动，请参考 <http://www.ftdichip.com/Products/ICs/FT232R.htm>，需要打开驱动程序的 VCP 功能，生成一个虚拟串口。

第二步是由串口终端软件 PuTTYtel (<http://www.putty.org> , Telnet-only client) 通过虚拟串口和 ESP8266 通讯，PuTTYtel 设置如下：

Serial line：虚拟串口

Speed：一般是 115200

Connection type: Serial

Open

断开 FLASH 开关，并按一下 RESET，终端会显示一些字符，最后一行应该是：

ready

首先用一个最简单的命令：AT <return> <ctrl-J>

终端回应：OK

然后测试 WiFi 连接:

AT+CWMODE=1 <return> <ctrl-J>, 终端回应: OK

AT+CWLAP 终端回应: OK, 终端回应: 找到的 WiFi 接点, 它们的加密方式, 名称, 信号强度以及 MAC 地址

AT+CWJAP=“接点名称”, “密码” <return> <ctrl-J>, 终端回应: OK

AT+CIFSR <return> <ctrl-J>, 终端回应:

+CIFSR:STAIP, “ESP8255 的 IP 地址”

+CIFSR:STAMAC, “ESP8255 的 MAC 地址”

ESP8255 从接点的 DHCP 服务中取得了 IP 地址证明连接成功。

更换 NodeMCU 软件

使用 NodeMCU, 简练的 LUA 代码就能够实现 IoT 功能。下面几个步骤将 ESP8266 原来的系统换成 NodeMCU 系统。

第一步是从 <https://nodemcu-build.com> 网站创建 NodeMCU 系统, 填入收取 NodeMCU 系统的邮件地址, 根据 8266 配置储存器的大小选择系统类型, 512KB 的储存器就能满足车库 IoT 控制的要求, 然后选择如下系统模块:

- | | | | |
|---|---|--|---|
| <input type="checkbox"/> ADC 🌐 | <input checked="" type="checkbox"/> file 🌐 | <input type="checkbox"/> PCM 🌐 | <input type="checkbox"/> struct 🌐 |
| <input type="checkbox"/> ADXL345 🌐 | <input type="checkbox"/> gdbstub 🌐 | <input type="checkbox"/> perf 🌐 | <input type="checkbox"/> Switec 🌐 |
| <input type="checkbox"/> AM2320 🌐 | <input checked="" type="checkbox"/> GPIO 🌐 | <input type="checkbox"/> PWM 🌐 | <input type="checkbox"/> TM1829 🌐 |
| <input type="checkbox"/> APA102 🌐 | <input type="checkbox"/> HMC5883L 🌐 | <input type="checkbox"/> RC (no docs) | <input checked="" type="checkbox"/> timer 🌐 |
| <input type="checkbox"/> bit 🌐 | <input type="checkbox"/> HTTP 🌐 | <input type="checkbox"/> rfswitch 🌐 | <input type="checkbox"/> TSL2561 🌐 |
| <input type="checkbox"/> BME280 🌐 | <input type="checkbox"/> HX711 🌐 | <input type="checkbox"/> rotary 🌐 | <input type="checkbox"/> U8G 🌐 |
| <input type="checkbox"/> BMP085 🌐 | <input type="checkbox"/> I ² C 🌐 | <input type="checkbox"/> RTC fifo 🌐 | <input checked="" type="checkbox"/> UART 🌐 |
| <input type="checkbox"/> CJSON 🌐 | <input type="checkbox"/> L3G4200D 🌐 | <input type="checkbox"/> RTC mem 🌐 | <input type="checkbox"/> UCG 🌐 |
| <input type="checkbox"/> CoAP 🌐 | <input type="checkbox"/> mDNS 🌐 | <input type="checkbox"/> RTC time 🌐 | <input type="checkbox"/> websocket 🌐 |
| <input type="checkbox"/> crypto 🌐 | <input type="checkbox"/> MQTT 🌐 | <input type="checkbox"/> Sigma-delta 🌐 | <input checked="" type="checkbox"/> WiFi 🌐 |
| <input type="checkbox"/> DHT 🌐 | <input checked="" type="checkbox"/> net 🌐 | <input type="checkbox"/> SNTP 🌐 | <input type="checkbox"/> WS2801 🌐 |
| <input type="checkbox"/> encoder 🌐 | <input checked="" type="checkbox"/> node 🌐 | <input type="checkbox"/> Somfy 🌐 | <input type="checkbox"/> WS2812 🌐 |
| <input type="checkbox"/> end user setup 🌐 | <input type="checkbox"/> 1-Wire 🌐 | <input type="checkbox"/> SPI 🌐 | |

最后启动 “Start your build”, 1 小时内, 需要的 NodeMCU 下载链接会送到信箱中, 这个应用不需要浮点运算, 下载 integer.bin 结尾的代码就可以。

第二步是将 NodeMCU 系统写入 ESP8266, 从 <https://github.com/nodemcu/nodemcu-flasher> 下载 WIN32 或者 WIN64 版本的 ESP8266Flasher.exe, 用 USB 电缆连接 FT232RL 到 PC, 闭合 FLASH 开关, 按一下 RESET, 运行 ESP8266Flasher.exe, 根据 ESP8266 的储存器设置好 “Advance” 中的选项 (请参考 ESP8266Flasher.exe 相关使用文档), 在 “Config” 中点击齿轮符号加载下载的 NodeMCU 系统代码, 最后在 “Operation” 中填入虚拟串口并启动 “Flash”, ESP8266Flasher.exe 将显示 ESP8266 的 MAC 地址, 等进度条填满后, 系统更新就完成了。

添加车库门控制代码

从<https://esp8266.ru/esplorer/>下载 ESplorer.zip, 解压缩到一个工作目录中, ESplorer 需要 JAVA SE V7 或者更新支持, 合适的 JAVA 运行库可以从<http://java.com/download> 下载并安装。

从本 GITHUB 中下载 init.lua 到任一目录, 运行 ESplorer.bat, 打开下载的 init.lua 程序, 必须要修改的是 WiFi 接点的名称和密码。

```
wifi.sta.config("SSID", "password")
```

开关车库门密码是 “if(pwd == "12345678") then” 中的 12345678, 可以改成自己需要的任何字符。

断开 FLASH 开关, 按一下 RESET, 在 ESplorer 右栏中选者虚拟串口和 115200 波特率, 点击 Open, 然后点击一下 “Chip ID” 确认通讯正常。

回到左栏, 先点击上方的 Save 盘符, 然后点击下方的 “Save&Compile”, 然后点击右栏中的 Reset, 等待几秒, 然后从右栏下方选择栏中选者 “wifi.sta.getip()”, 右栏终端窗口会显示 ESP8266 的 IP 地址, 然后用一网页浏览器打开这个 IP 地址, 就能见到本文档首页中的控制页面。

测试和安装

测试

将写好程序的 ESP8266 接入按车库控制器原理图建造的电路板中, 接入合适电源, 几秒钟后打开控制网页, 填入密码, 按动 “车库门”, 会听到继电器快关的声音, 三分钟后, 会再次听到继电器快关的声音, 电击开灯/关灯按键, 会听到继电器快关的声音。

安装

将 Relay1 并联到车库门手动开关, Relay2 控制车库灯可选, 如果灯是市电或者其他高电压类型的, 请小心操作避免危险, 最后接入合适电源, 重复上面测试, 观察车库门和灯的控制效果。