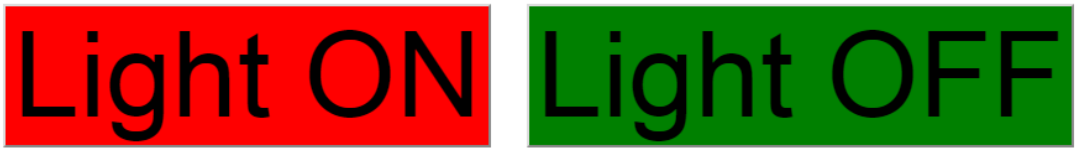
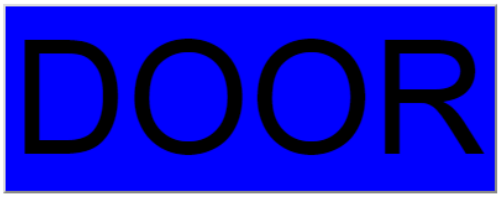
Web Controlled Garage Door and Light

Background

Most electric garage door opener is controlled by a push button, pressing once to open the door and pressing again to close the door. The web controller drives a relay which is parallel to the push button. Pushing the virtual button on the web page activates the relay for a predefined time of 0.5s, emulating the manual activation of the push button. There is a password input on the web page. Only the correct password can activate the relay. There is also a light control relay. The light relay cn also be turn on or off by 2 virtual button the web page. If the lioght relay is off, activating the door relay also turn on light relay for another predefined time of 3 minutes. The light virtual button press will cancel timed turning on activated by door relay.

The web controller is designed using ESP8266 running NodeMCU. The sample LUA code shows how to do WiFi connection, web page with virtual buttons, HTTP GET/POST data processing and relay timing.

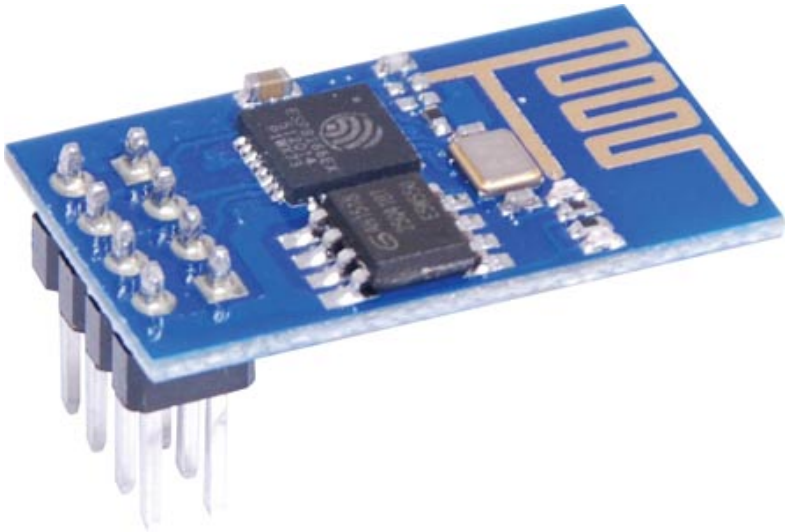




Controller Web Page

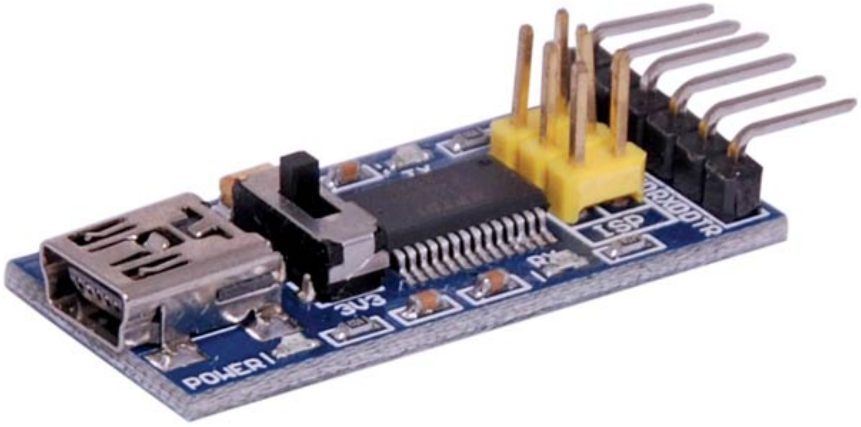
Hardware

ESP8266 WiFi Module



ESP8266 WiFi module can be purchased from TaoBao(Chinese), EBay, or some local electronic retailers in Australia, ie. www.altronics.com.au.

FT232RL USU-UART Interface

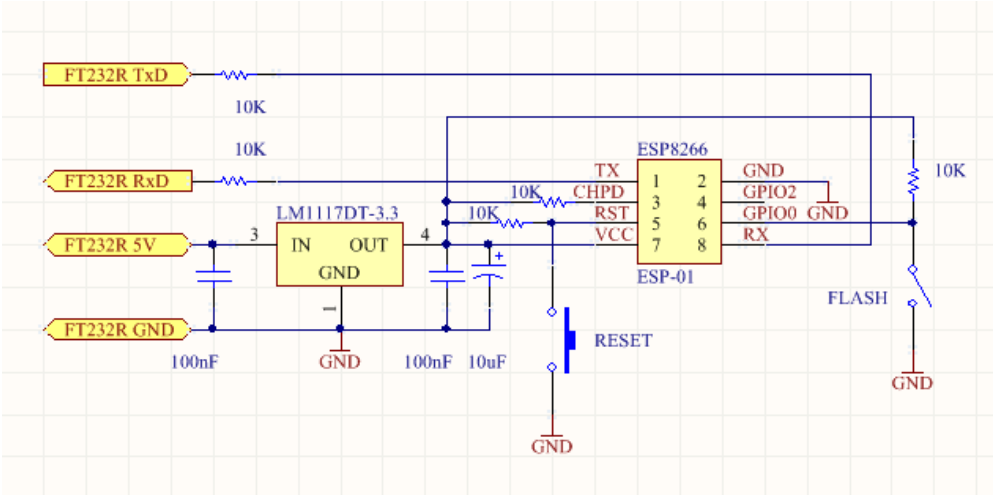


It is used as ESP8266 programmer and debugger. Like ESP8266, it can be purchased from TaoBao(Chinese),EBay, or or some local electronic retailers in Australia, ie. www.altronics.com.au. The datasheet of the USB-UART interface chip can be download from:

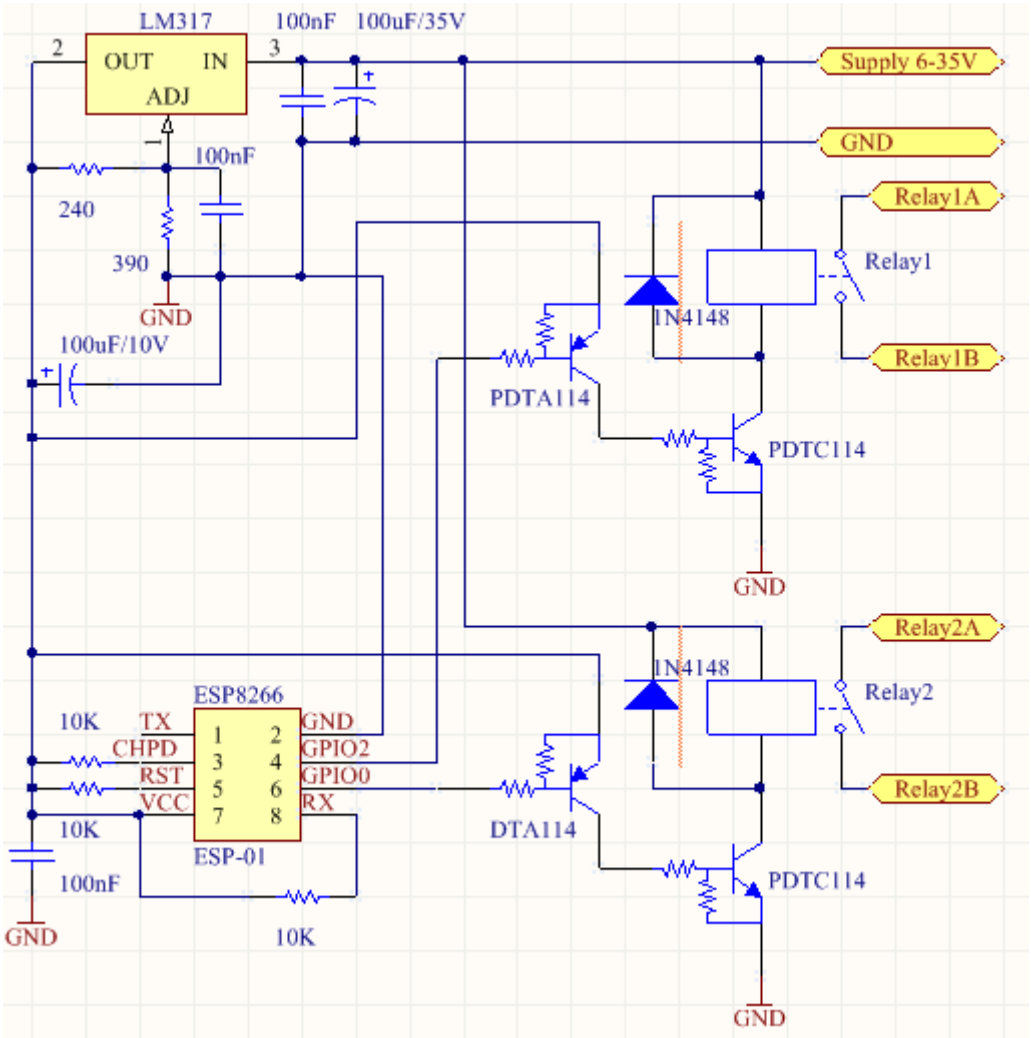
http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232R.pdf

Circuit Schematics

The schematics for the programmer/debugger and web controller are shown as the below.



FT232RL Interface connected to ESP8266



Web Contrller

The supply voltage of the Relay1 and Relay2 need to match the supply voltage of the web controller. Many electric garage door openers use 24V DC. Relay1 is for door activation and relay2 is for light.

Software

Test FT232RL connection

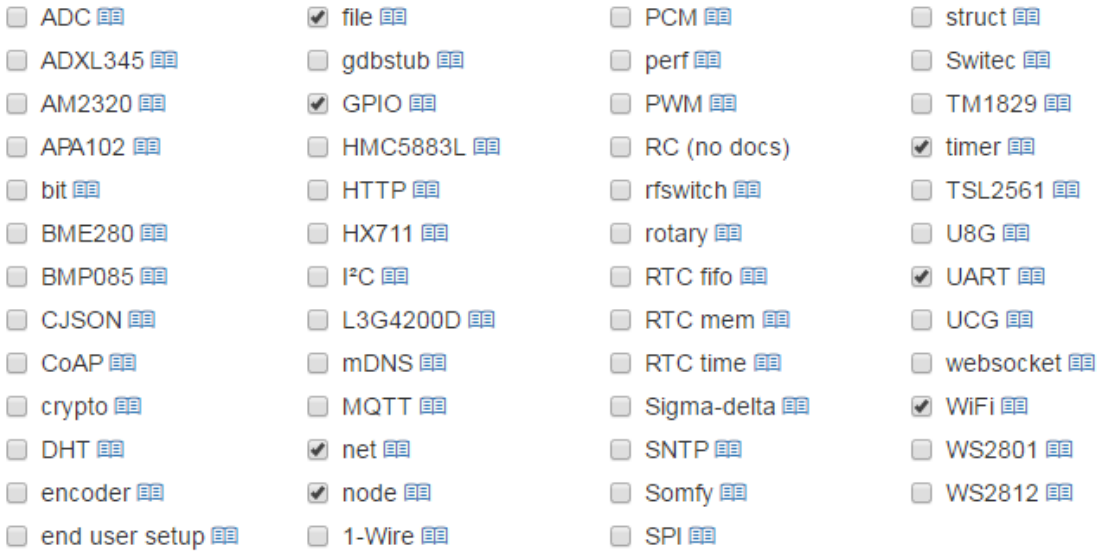
This test verifies the FT232RL Interface connected to ESP8266

1. Connect FT232RL to a Windows PC using a USB cable and install FT232RL driver, see http://www.ftdichip.com/Products/ICs/FT232R.htm. The VCP function of the driver needs to be activated for get a virtual com port.
2. Download PuTTYtel from http://www.putty.org. Run it and connect PC to ESP8266.

   The Setting of PuTTYtel:
   Serial line: the virtual com port
   Speed: 115200 normally
   Connection type: Serial
3. Then press open button.
4. Keep FLASH switch open and press RESET. The PuTTY terminal shows some rubbish characters but the last line should be "ready".
5. Try first command. Type AT<return><ctrl-J>. ESP8266 should respond OK.
6. Try WiFi connection command. Type AT+CWMODE=1<return><ctrl-J> and ESP8266 should respond OK. Type AT+CWLAP<return><ctrl-J> and ESP8266 should list all WiFi access points in its range, with their SSID, encryption, signal strength and MAC. Type AT+CWJAP="SSID","password"<return><ctrl-J> and ESP8266 should respond OK. Type AT+CIFSR<return><ctrl-J> and ESP8266 should respond +CIFSR:STAIP,"ESP8266 IP address" and +CIFSR:STAMAC,"ESP826 MAC". This approves the connection to the WiFi access point and assignment of IP address using DHCP.

Flash NodeMCU firmware

1. Get customer built NodeMCU firmware from https://nodemcu-build.com. Fill in the receiving email address. Select firmware type according to ESP8266 on board flash memory size. 512KB flash memory is enough for the web controller. Choose minimum module shown as the below.



Press "Start your build', NodeMCU customer built firmware download links should be sent to the email address provided. The web controller doesn't need float point. download the link ending with "integer.bin" is enough.

2. Download ESP8266Flasher.exe from https://github.com/nodemcu/nodemcu-flasher. Set FLASH switch close. Press RESET. Run ESP8266Flasher.exe. Set "Advance" option according to ESP8266 flash memory (see ESP8266Flasher.exe user manual). Press gear shape button in "Config" tag to load downloaded NodeMCU firmware. Then fill the virtual com port in "Operation" tag and press "Flash" button. ESP8266Flasher.exe should shows ESP8266 MAC and progress of flashing.

Web Controller LUA Code

1. Download JAVA SE7 or newer from http://java.com/download and install. Download ESplorer.zip from https://esp8266.ru/esplorer/. Unzip it to a working directory.
2. Download init.lua from https://github.com/cookhome/IoT/tree/master/ESP8266%20Garage%20Door%20Controller. Run ESplorer.bat and load init.lua. Modify line wifi.sta.config("SSID","password") with the SSID and password of the router. Modify if(pwd == "12345678")then with the door activation password (replacing 12345678).
3. Set FLASH switch open, press RESET. Fill in the virtual com port and 115200 as the baud rate on the right side of the ESplorer window. Press Open button and Chip ID button to verify the communication to ESP8266.
4. Press Save and Save&Compile button on the left side window. Press Reset button on the right side window. Wait a few seconds for the WiFi connection. The choose wifi.sta.getip() from the pull down selection. The terminal windows show ESP8266's IP address. Using a web browser to view the IP address. The web controller should show up. To fix IP address, LUA code can be modified. It can also be done with the router by assign a fixed IP to a predefined MAC address.

Arduino Based Firmware

Although NodeMCU is easy to get started, LUA is very slow running comparing to compiled C language. Although C can be used to write NodeMCU module, it is relatively complex without IDE support. ESP8266 Arduino provided an IDE environment for easier C language development.

Arduino Install and Setup for ESP8266

Download and install latest version of Arduino IDE from "https://www.arduino.cc/en/Main/Software".

Run installed Arduino IDE, fill "http://arduino.esp8266.com/stable/package_esp8266com_index.json" in option "Additional Boards Manager URLs" under File/Preference menu item. Then open menu item "Tools/Board/Board Manager...". Click on " esp8266 by ESP8266 Community" and Install button. Waiting package to be installed.

Return "Tools/Board" meni item. select "Generic ESP8266 Module" from the right side list. Set:
Flash Mode:"QIO"
Flash Frequency:"40MHz"
CPU Frequency:"80MHz"
Flash Size:"512K (64K SPIFFS)
Debug Port: "Disabled"
Debug Level:"None"
Reset Method:"ck"
Upload Speed:"115200"
Port:"FT232 COM Port"

Compile, Download and Run

Copy all code GarageDoorCtrl.ino from GitHub to Arduino IDE editor. Modify WiFi connection SSID and password. Then change garage door open/close password "12345678" to anything easy to remember. Save as a new project. Close Flash switch and press RESET button. Click on Arduino IDE Sketch/Upload menu item. Waiting compiling and downloading to be finished. The code run automatically after downloading regardless of Flash switch state.

The IP address of the controller can be obtained from routers' DHCP list. PuTTYtel can be used to receive ESP8266 serial output.

Testing

Plug the ESP8266 to the web controller circuit and power the circuit up. Browse the web page in a few seconds. Fill in door activation password and press DOOR button. Relay click sound should be heard. 3 minutes later, another relay click sound should be heard. Click ON and OFF, relay click sound should be heard.

Now the web controller is ready to be attached to the garage door opener.