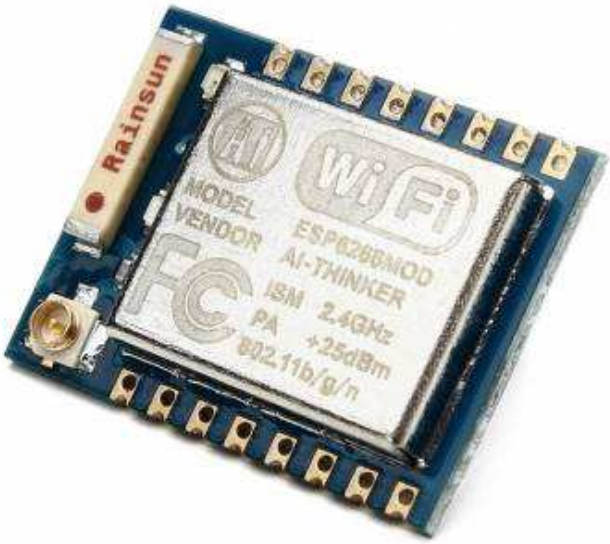
ESP8266 Bootloader Modes and GPIO state on Startup

Introduction

The ESP8266 (https://nurdspace.nl/ESP8266) is an extremely cheap wifi module with a fairly capable processor on board. Recently, it's exploded on the hobbyist scene due to its low cost. Even better, there's a high level development platform available (NodeMCU (https://github.com/nodemcu/nodemcu-firmware)) which runs eLua code making programming quick and simple.

We've been using the ESP-07 (http://l0l.org.uk/2014/12/esp8266-modules-hardware-guide-gotta-catch-em-all/) module in a lot of our projects recently which breaks out nine of the on-board GPIO pins. Although the pin pitch is a non-standard 2mm, breakout boards are available and the smaller size is useful for PCB projects.



About this post

Posted on: Tuesday, 2nd of June 2015

Posted by: Richard Webb

Tagged: ESP8266 (/tags/esp8266), bootloader (/tags/bootloader), ESP-07 (/tags/esp-07), IoT (/tags/iot), electronics (/tags/electronics)

Recent posts

A Serial Communication Library for Arduinos (/tech-blog/serial-communication-library-arduinos) - *12th Dec 2015*

SWmobile talk

Documentation for the ESP8266 is fairly sparse, at least in any official format, but there is a large amount of discussion in various places, most notably the ESP8266 forum (http://esp8266.com). One of the stumbling blocks we encountered was that on startup, the module can enter a number of bootloader modes depending on GPIO pin states. This means that if you want to use any of those pins, you have to be quite careful.

Flashing the NodeMCU firmware

To flash NodeMCU (or any other firmware) you'll need to connect the following pins:

- GPIO 0: LOW
- GPIO 2: HIGH
- GPIO 15: LOW

Apply 3.3V and GND and use a 3.3V UART to connect the device to a computer. We tend to use esptool.py (https://github.com/themadinventor/esptool/blob/master/esptool.py) to actually do the flashing. Hackaday has a nice guide to connecting everything here (http://hackaday.com/2015/03/18/how-to-directly-program-an-inexpensive-esp8266-wifi-module/)

If, like us, you're using the ESP-07 and you need to flash a lot of them, it's fairly simple to make a jig with Pogo Pins where you can clamp the module during flashing.

You can generally find pre-built versions of NodeMCU around github but if your application uses a lot of memory, you'd do well to remove some of the unneeded modules in user_modules.h (https://github.com/nodemcu/nodemcu-firmware/blob/master/app/include/user_modules.h) and rebuild the binary.

Bootloader Modes

The bootloader can go into a number of modes depending on the state of GPIOs 0, 2 and 15. The two useful modes are the UART download mode (for flashing new firmware) and the flash startup mode (which boots from flash).

	GPIO 0	**GPIO 2**	**GPIO 15**
UART Download Mode (Programming)	0	1	0
Flash Startup (Normal)	1	1	0

(/tech-blog/swmobile-talk) *- 11th Nov 2015*

Really simple MQTT IoT logging with Keen IO (/tech-blog/simple-mqtt-iot-logging) *- 7nd Jul 2015*

ESP8266 Bootloader Modes and GPIO state on Startup (/tech-blog/esp8266-bootloader-modes-and-gpio-state-startup) *- 6nd Jun 2015*

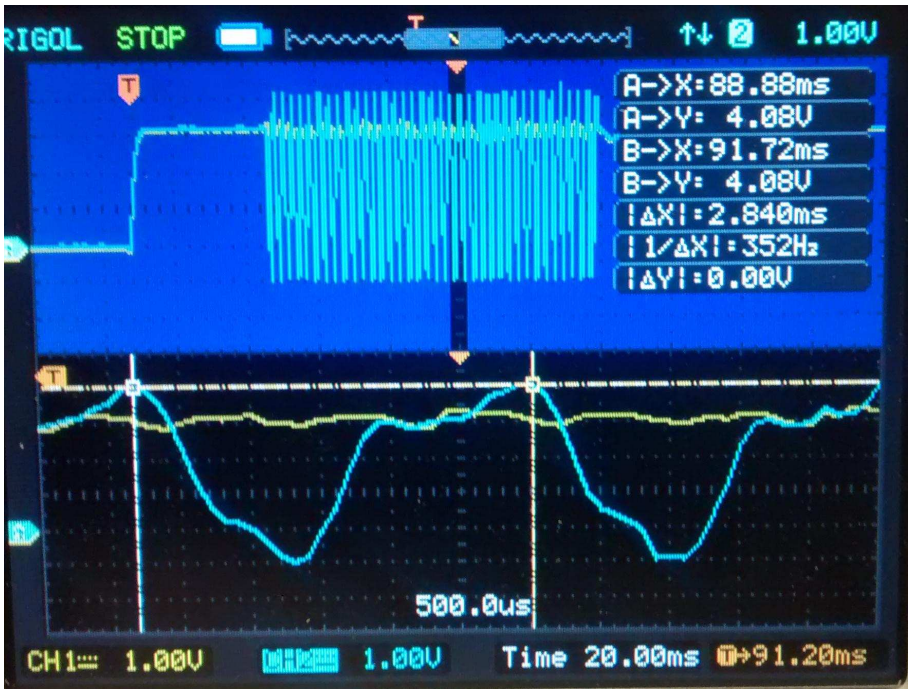
A Brief, but Practical Introduction to the MQTT Protocol and its Application to IoT (/tech-blog/brief-practical-introduction-mqtt-protocol-and-its-application-iot) *- 6nd Jun 2015*

SD-Card Boot	0	0	1

GPIO state on startup

When choosing GPIO pins to use, it's best to avoid GPIO 0, 2 and 15 unless absolutely necessary. If you do end up using them, you'll need pullups / pulldowns to ensure the correct bootloader mode. You should also be aware of the fact that GPIO 0 is driven as an output during startup (at least with NodeMCU).

Here's what we found: 40ms after startup, the GPIO0 line is driven with a signal at around 350 Hz for around 100 ms. So make sure you don't rely on GPIO0 being stable for the first ~200 ms after startup.



(https://www.linkedin.com/company/zoetrope-limited)

Zoetrope 3D product photography (https://zoetrope-imaging.co.uk/)

/favorite?tweet_id=590428128343941120)

Company No: 08742496

FROM THE BLOG

A Serial Communication Library for Arduinos (/tech-blog/serial-communication-library-arduinos)
Recently, we've been working on a large internet connected machine. This involves quite a large amount of electronics to drive all it's motors,...

SWmobile talk (/tech-blog/swmobile-talk)
Thanks for having me SWmobile! I've uploaded a web version of my deck, which you should be able to see below: