Có nhiều loại cáp ( module) chuyển đổi loại này, tuy nhiên khi chọn module chuyển đổi USB-Serial để nạp code cho ESP8266 chúng ta cần chú ý những điều sau:

* Điện áp của các IO pin của mạch không vượt quá 3.3V, nếu sử dụng loại module có điện áp IO 5V sẽ phá hỏng module của các bạn.
* Module USB-Serial nên có pin CTS để khi upload code được dễ dàng.

Các module đã được test và làm việc tốt:

* FT232RL
* CP2102
* CH340G

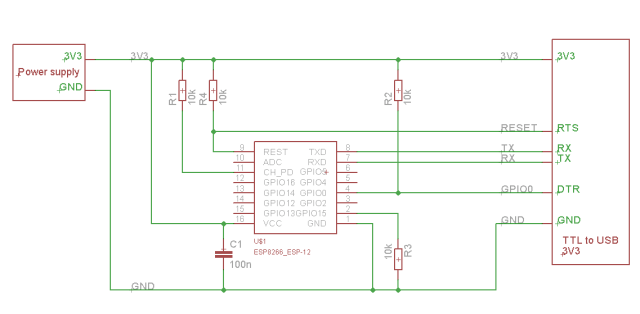
Mạch nạp cho esp8266

Hiện nay trên các board đã liệt kê ở bài viết trước đều sử dụng button để vào bootloader mode, điều này khá bất tiện khi chạy test code, mỗi lần upload đều phải nhấn button reset sau đó nhấn giữ button gpio0, nếu chạy test code liên tục thì đúng là ác mộng. Dưới đây là sơ độ nguyên lý được đưa ra bới team phát triển Arduino-ESP8266

| **PIN** | **Resistor** | **Serial Adapter** |
| --- | --- | --- |
| VCC |  | VCC (3.3V) |
| GND |  | GND |
| TX or GPIO2\* |  | RX |
| RX |  | TX |
| GPIO0 | PullUp | DTR |
| Reset\* | PullUp | RTS |
| GPIO15\* | PullDown |  |
| CH\_PD | PullUp |  |

* Chú ý
  + GPIO15 còn có tên là MTDO
  + Reset còn có tên là RSBT or REST, thêm điện trở PullUp để hoạt động ổn định
  + GPIO2 thay thế cho TX khi đang ở boot loader mode
  + Nên mắc thêm tụ điện 470-1000µF giữa VCC và GND để tránh bị sụt áp dẫn tới khởi động lại ESP8266 khi nó gửi lượng dữ liệu lớn qua wifi

ESP to Serial



Phần cứng tối thiểu cho Bootloading:

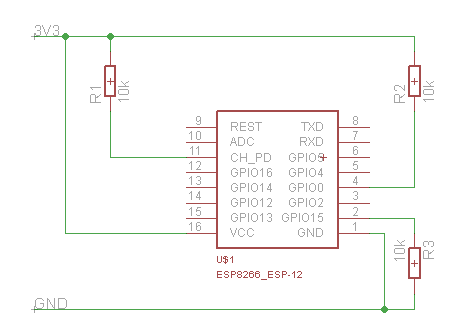
ESPxx Hardware

| **PIN** | **Resistor** | **Serial Adapter** |
| --- | --- | --- |
| VCC |  | VCC (3.3V) |
| GND |  | GND |
| TX or GPIO2 |  | RX |
| RX |  | TX |
| GPIO0 |  | GND |
| Reset |  | RTS\* |
| GPIO15 | PullDown |  |
| CH\_PD | PullUp |  |

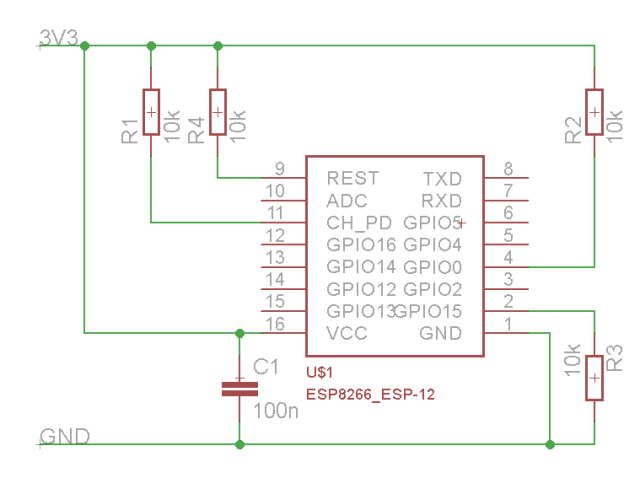
* Chú ý: nếu không có PIN RTS  ở USB-Serial thì cần khởi động lại bằng tay.

Phần cứng tối thiểu cho ESP8266 hoạt động tốt

| **PIN** | **Resistor** | **Power supply** |
| --- | --- | --- |
| VCC |  | VCC (3.3V) |
| GND |  | GND |
| GPIO0 | PullUp |  |
| GPIO15 | PullDown |  |
| CH\_PD | PullUp |  |



Phần cứng để ESP8266 họat động ổn định nhất:



Boot Messages và các Modes

Khi ESP module khới động, nó sẽ kiểm tra trạng thái của các Pins 0, 2 và 15 để vào các boot mode:

| **GPIO15** | **GPIO0** | **GPIO2** | **Mode** |
| --- | --- | --- | --- |
| 0V | 0V | 3.3V | Uart Bootloader (vào bootloader để nạp code) |
| 0V | 3.3V | 3.3V | Boot chương trình đã được nạp vào chip(SPI flash) |
| 3.3V | x | x | SDIO mode (không sử dụng khi lâpị trình với Arduino) |

Khi ESP8266 khởi động sẽ có một thông báo gửi đến cổng UART (boot messages), dịch thông báo này sẽ giúp ích cho chúng ta rất nhiều cho việc debug cũng như tìm ra một phần nguyên nhân các lỗi phát sinh trong quá trình chạy. Thông thường thông điệp gửi đến UART có dạng:

*rst cause:2, boot mode:(3,6)*

rst cause

| **Number** | **Description** |
| --- | --- |
| 0 | unknown |
| 1 | normal boot |
| 2 | reset pin |
| 3 | software reset |
| 4 | watchdog reset |

boot mode

| **Number** | **GPIO15** | **GPIO0** | **GPIO2** | **Mode** |
| --- | --- | --- | --- | --- |
| 0 | 0V | 0V | 0V | Not valid |
| 1 | 0V | 0V | 3.3V | Uart |
| 2 | 0V | 3.3V | 0V | Not valid |
| 3 | 0V | 3.3V | 3.3V | Flash |
| 4 | 3.3V | 0V | 0V | SDIO |
| 5 | 3.3V | 0V | 3.3V | SDIO |
| 6 | 3.3V | 3.3V | 0V | SDIO |
| 7 | 3.3V | 3.3V | 3.3V | SDIO |

với: number = ((GPIO15 << 2) | (GPIO0 << 1) | GPIO2);

[xyz-ihs snippet=”Adnow5x1″]

[xyz-ihs snippet=”Adnow1x5Mobile”]

<https://www.youtube.com/watch?v=37KlzYhJaQs&fbclid=IwAR00VipiZMSYLnJFn_uxnI9odZOzgSRe9atnw07vToHi5C8ioUn_camjwCs>