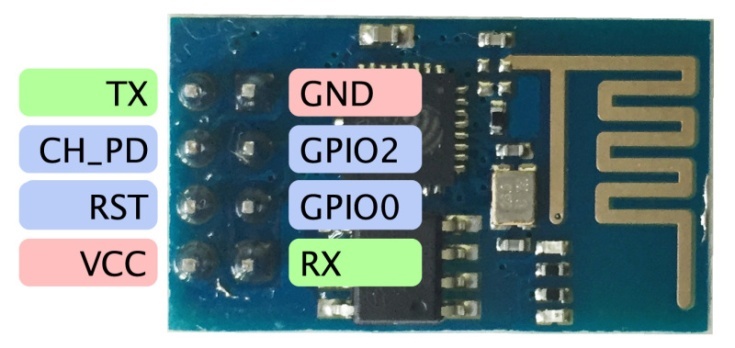
**ESP8266**



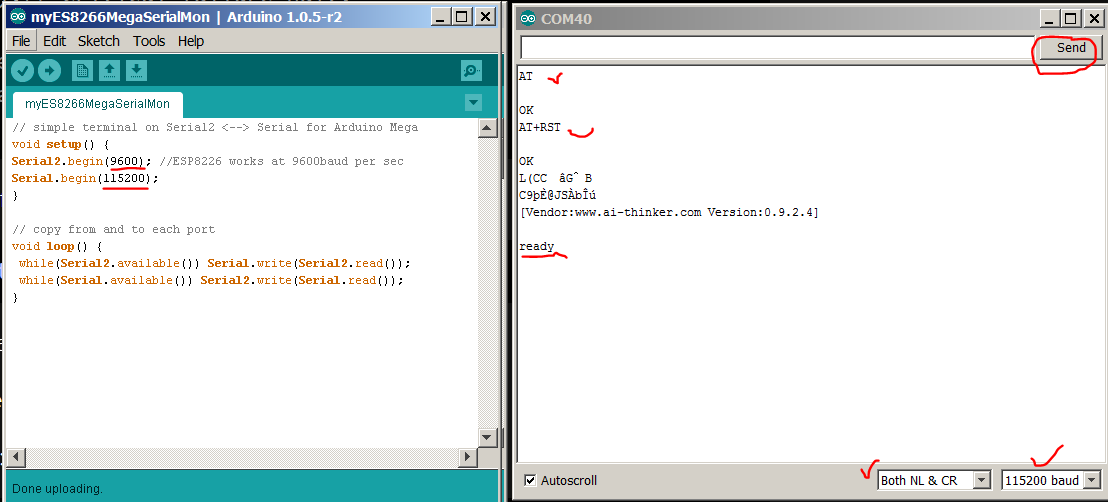
Engineered for mobile devices, wearable electronics and the Internet of Things (IoT) applications

* TX – transmission -> **ESP TX-> Arduino RX**
* GND – ground
* RST – reset – connected to 3.3v for logic level high
* VCC – 3v
* RX – receive -> **ESP RX-> Arduino TX**
* CH\_PD – chip power down – connected to 3.3v for logic level high

Mega has 4x serial communication -> serial port, 19 (RX) and 18 (TX), pins 17 (RX) and 16 (TX), pins 15 (RX) and 14 (TX) here we will use 16 and 17 pins for serial communication of esp8266 with arduino mega. TX(#16), RX (#17)

Note: baudrate must match on both ends of a pair of comms device

baudrate -> 115200 ---- both NL and CR



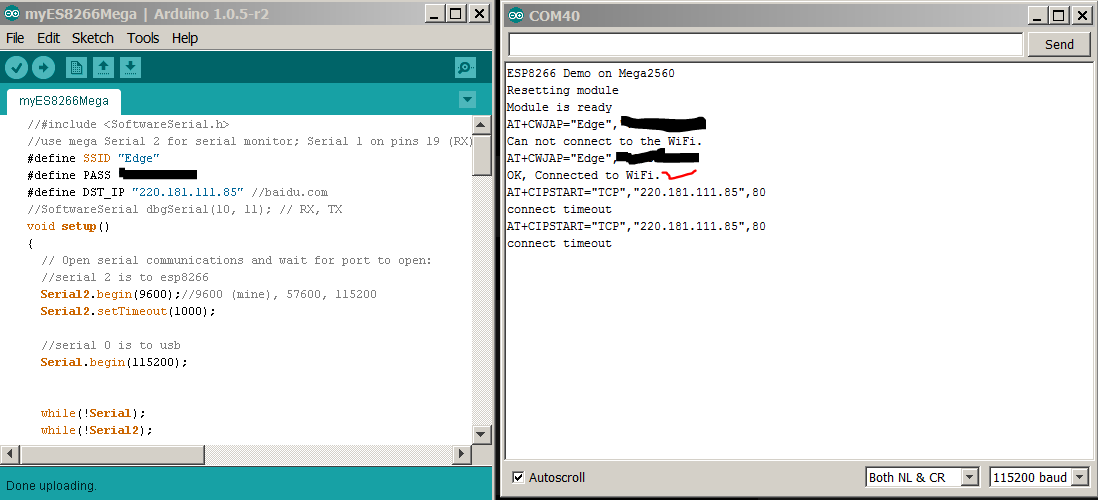
from the serial monitor window, enter **AT** and then press "send"; assuming all parameters are set accordingly, an "**OK**" will be replied by ESP8266.

to reset the ESP8266, enter **AT+RST** and then press "send".

some random data will then appear, follow by "ready"

Source code for testing ESP9266 on an Arduino with Internet

An wireless router/AP is setup with the SSID "Edge", Security is set as "WEP", and DHCP. The WAN port of this AP is connected to the Internet. In the following diagram ESP8266 is connected to the AP, and issued with a private IP. However, the WAN port on the AP does not have an IP assigned, hence there is no Internet access.



Few important points to note:

1. the baudrate in the blog post is 57600 whereas the ESP82665 on hand is 9600, with no mentioned or whatsoever w.r.t it in the blogpost of the manufacturer.
2. the CH\_PD & RESET are not held at logic high in the blogpost of the manufacturer.
3. open serial monitor and press reset on the arduino does not return any data at serial monitor. The main cause of frustration here is to figure out the no data displayed is caused by hardware e.g tx-rx pairs wiring, 3.3v and 5v circuitry wiring, etc or caused by software e.g baudrate on esp8266/usb or wrong code uploaded. The matter of fact, the statement itself is misleading. Assuming each and every parameter is config properly, there will be no data after performing the ritual of "open serial monitor and press the reset button on arduino" the magic lies at typing the command "AT" and then press "send" then the data will appear at output.

**Methodology & Observations: Setup(s) that does not work/ does work**

MCU: uno  
comm: serial0 only  
code: "Examples->basic->BareMinimum"  
en: 3.3v to ch\_pd and REST  
cct: voltage divider  
baud s0: 9600  
D0 rx-> tx ESP8266  
D1 tx-> RX ESP8266  
Serial mon on Serial0; AT and then "send" => no response  
//-----------------------  
MCU: uno  
comm: serial0 only  
code: "Examples->basic->BareMinimum"  
en: 3.3v to ch\_pd and REST  
cct: voltage divider  
baud s0: 9600  
D0 rx-> tx ESP8266  
D1 tx-> RX ESP8266  
SSCom32 on Serial0; AT and then "send" => no response  
//-----------------------  
MCU: uno  
comm: serial0 only  
code: "serial0 setup with 9600"  
en: 3.3v to ch\_pd and REST  
cct: voltage divider  
baud s0: 9600  
D0 rx-> tx ESP8266  
D1 tx-> RX ESP8266  
SSCom32 on Serial0; AT and then "send" => no response  
//-----------------------  
  
MCU: uno  
comm: serial0 only  
code: "serial0 setup with 9600"  
en: 3.3v to ch\_pd and REST  
cct: voltage divider  
baud s0: 9600  
D0 rx-> tx ESP8266  
D1 tx-> RX ESP8266  
Serial mon on Serial0; AT and then "send" => no response  
//-----------------------  
  
MCU: uno  
comm: serial0 only  
code: "modified hello word with RGB indicator"  
en: 3.3v to ch\_pd and REST  
cct: voltage divider  
baud s0: 9600  
D0 rx-> tx ESP8266  
D1 tx-> RX ESP8266  
Serial mon on Serial0 can't used  
  
//-----------------------  
MCU: mega  
comm: serial0 only  
code: "Examples->basic->BareMinimum"  
en: 3.3v to ch\_pd and REST  
cct: voltage divider  
baud s0: 9600  
D0 rx-> tx ESP8266  
D1 tx-> RX ESP8266  
Serial mon on Serial0; AT and then "send" => no response  
//--------------------------  
MCU: mega  
comm: serial0 & serial2 for ESP8266  
code: "copy s0<->s2"  
en: 3.3v to ch\_pd and REST  
cct: voltage divider  
baud s0: 115200  
baud s2: 9600  
D0 rx-> tx ESP8266  
D1 tx-> RX ESP8266  
Serial mon on Serial0; AT and then "send" => OK; AT+RST and then "send" => ready;

=========================================================