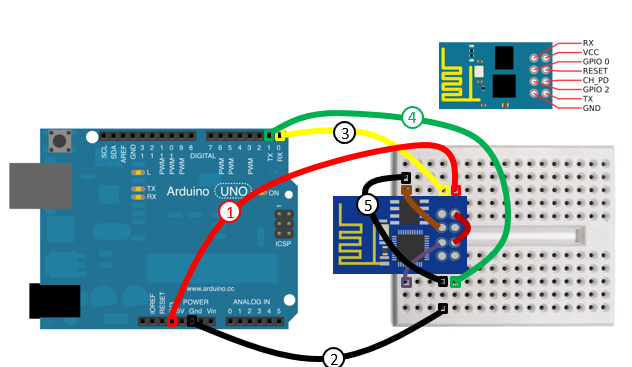
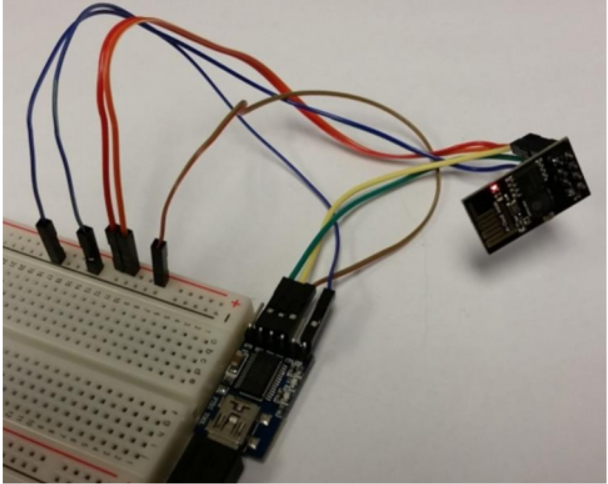
Getting Started With ESP8266

Supplies:

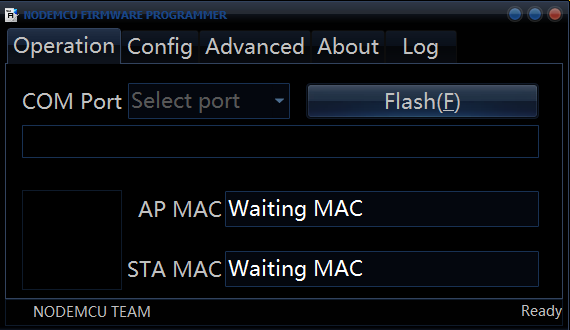
* Arduino Uno with a 3.3V power pinout
* ESP8266 wifi module
* Male and Female connector pins
* ESPlorer
  + http://esp8266.ru/esplorer-latest/?f=ESPlorer.zip
* NodeMCU Flasher
  + <https://github.com/nodemcu/nodemcu-flasher/raw/master/Win32/Release/ESP8266Flasher.exe>
* Computer

Below is a diagram of how the ESP8266 will be wired to the Arduino to flash the firmware. The chip will not fit in a standard size breadboard shown in the diagram, so use connector pins like the second image.





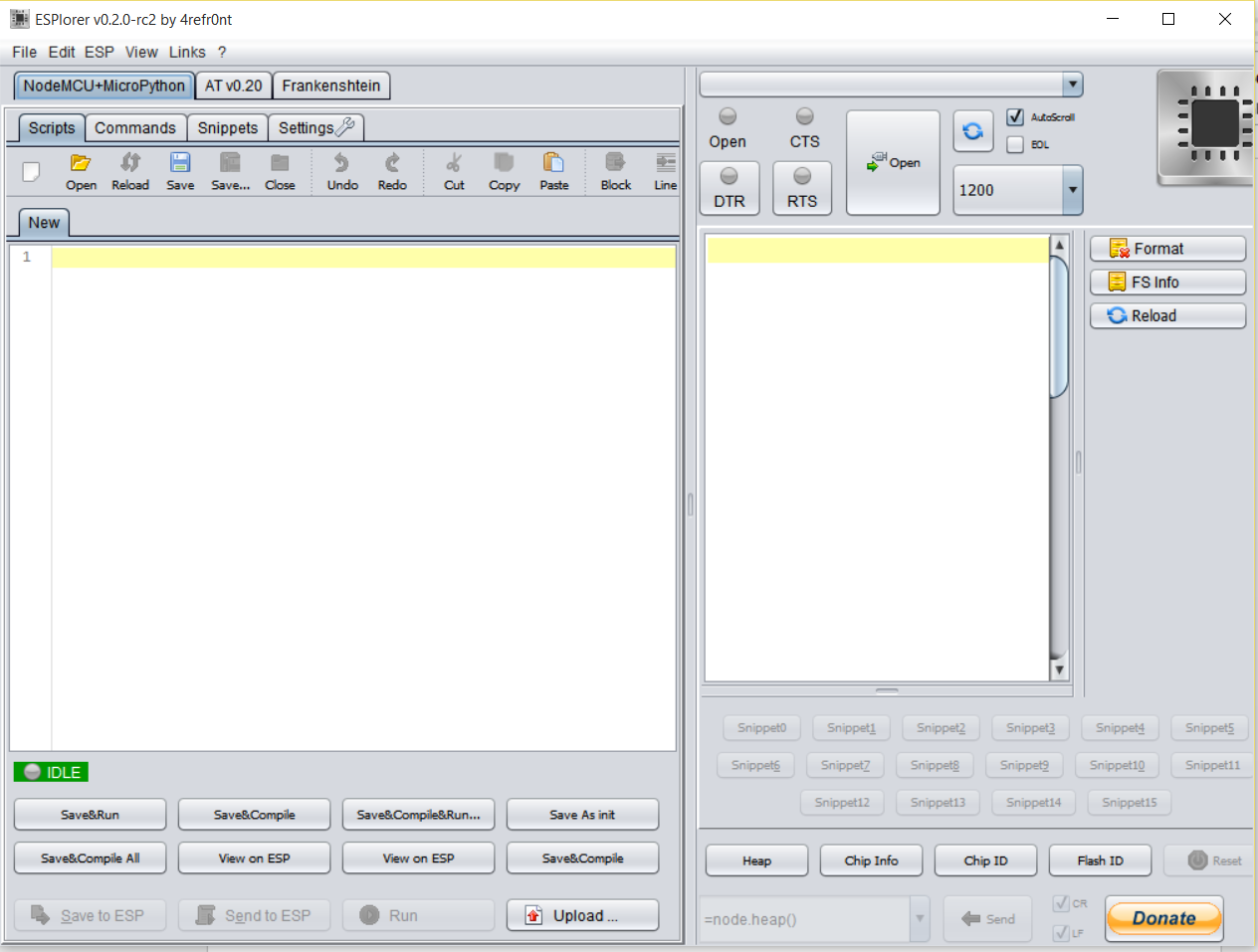
Once the connections are set, plug Arduino into computer and open NodeMCU flasher (shown below). **NOTE: NodeMCU only works on Windows 7.** Find the COM port the Arduino is located on and press flash. When you see a green circle with a check mark, you know the firmware has been successfully installed.



After you have installed the firmware, remove the GPIO 0 pin from ground and put it to power with Vcc and CH\_PD on the bread board. The next step is to put code onto the ESP8266 which will allow it to act as a webpage. To do this, download ESPlorer. It can be found at this link:

<http://esp8266.ru/esplorer-latest/?f=ESPlorer.zip>

Unzip the file and save. Run the JAR executable file, and this window should come up:



After opening the program, make sure you are under the NodeMCU+MicroPython tab and the scripts tab. Next go to the right hand panel and use the drop down menu to select the COM port your Arduino is on. Set the BAUD rate to 9600 and press open to open the connection.

An easy way to make sure everything is working is to press either the Chip ID or the Chip Info buttons on the bottom right hand side. Next, download this file: <http://randomnerdtutorials.com/wp-content/uploads/2015/02/init.zip> and unzip. Go to ESPlorer, select Open and find the init.lua file. Make sure that your file’s name is init.lua. On line two of the code, make sure to enter the wifi name and your password.

Now that everything is all set up, go to ESPlorer and find the drop down menu of commands towards the bottom next to the Send button. Select the “wifi.sta.getip()” command and press send. Now take the IP address that shows up and type it into your browser after “http://”. The webpage should appear.



Now that your ESP8266 is ready, we have to upload some code to our Arduino so it can receive information from the wifi module. When uploading code, it is important to disconnect the TX and RX pins from the Arduino or else the code will be uploaded to the wifi chip and you will have to re-upload the LUA code. Upload the following code:

//start code

int gpio0 = 2;

int gpio1 = 3;

int valOne = 0;

int valZero = 0;

void setup() {

// put your setup code here, to run once:

Serial.begin(9600);

pinMode(gpio0, INPUT);

pinMode(gpio1,INPUT);

}

void loop() {

// put your main code here, to run repeatedly:

valZero = digitalRead(gpio0);

Serial.print("GPIO 0:");

Serial.print(valZero);

Serial.println("");

delay(500);

valOne = digitalRead(gpio1);

Serial.print("GPIO 1:");

Serial.print(valOne);

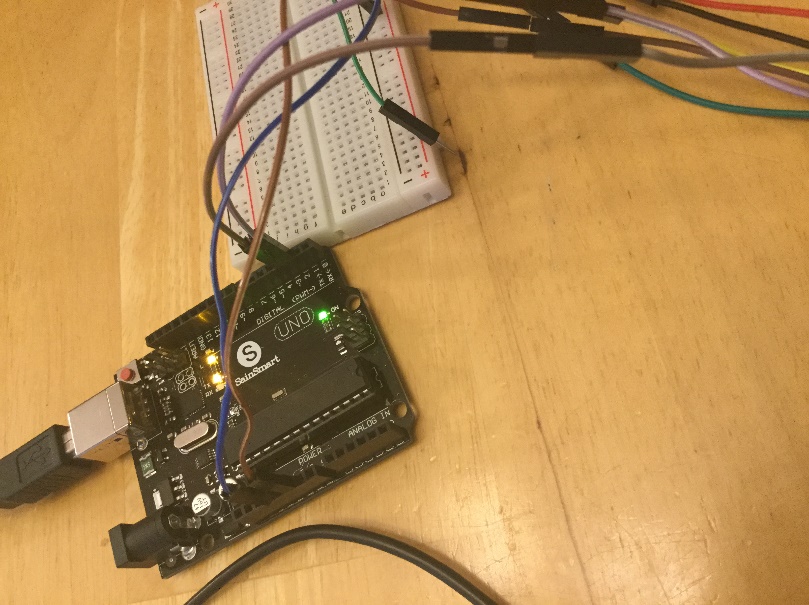
Serial.println("");

delay(500);

}

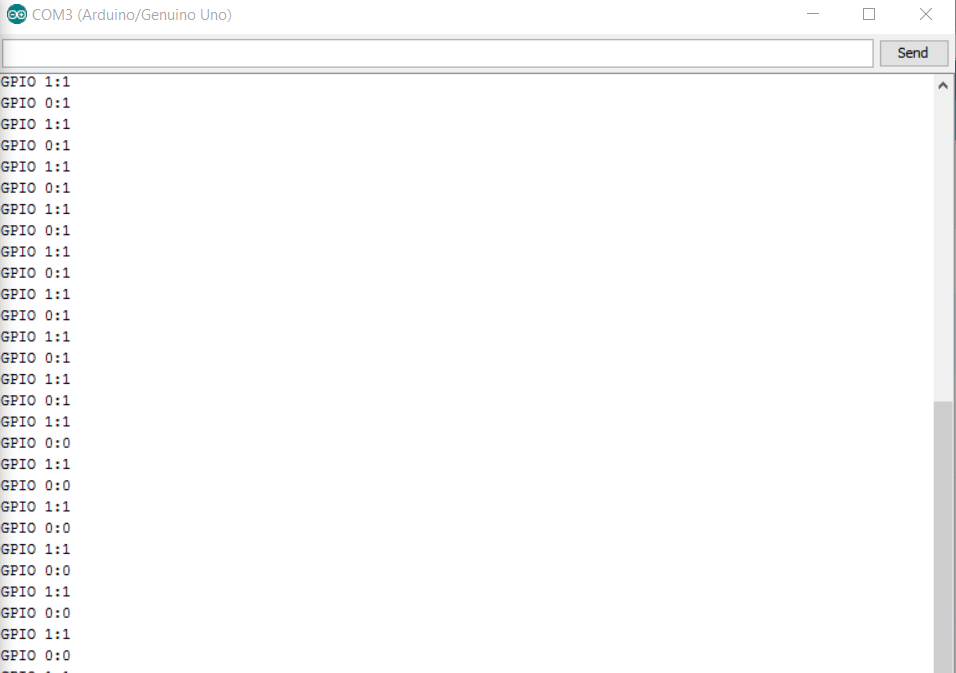
//end code

And here is what the hardware setup should look like:



The ESP8266 should have Vcc and CH\_PD pins connected to 3.3V power, ground to ground, the GPIO 0 pin connected to digital pin 2 on the Arduino and the GPIO 1 pin connected to digital pin 3.

Now that all the parts are in place, open up the webpage you just created. Also open up Arduino’s IDE and under tools select Serial Monitor. Now you should be able to press the On and Off buttons on the web page and watch the zeros and ones change on the serial monitor.



From here, you can do just about anything with enough GPIO pins. If you create more buttons and implement more code and use output pins, you can control almost anything from your web page.