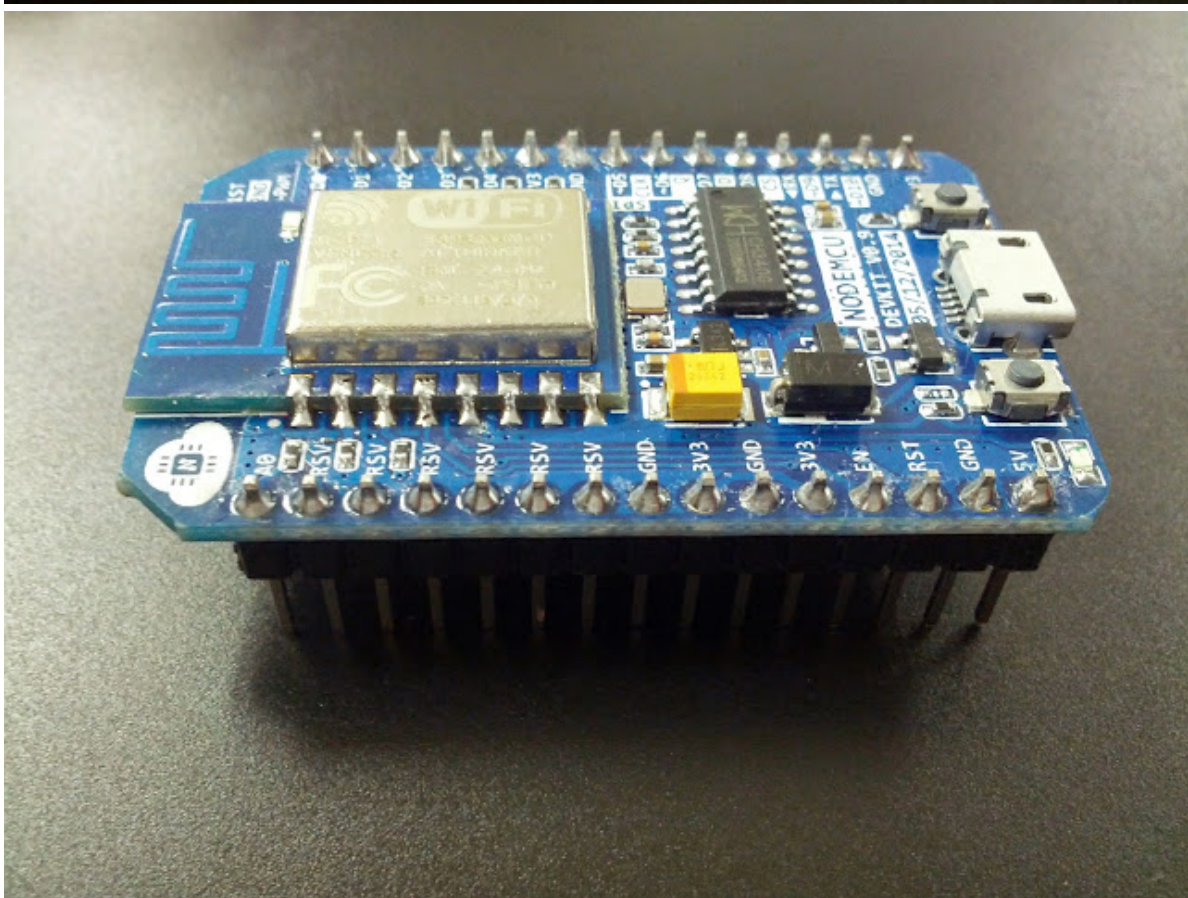
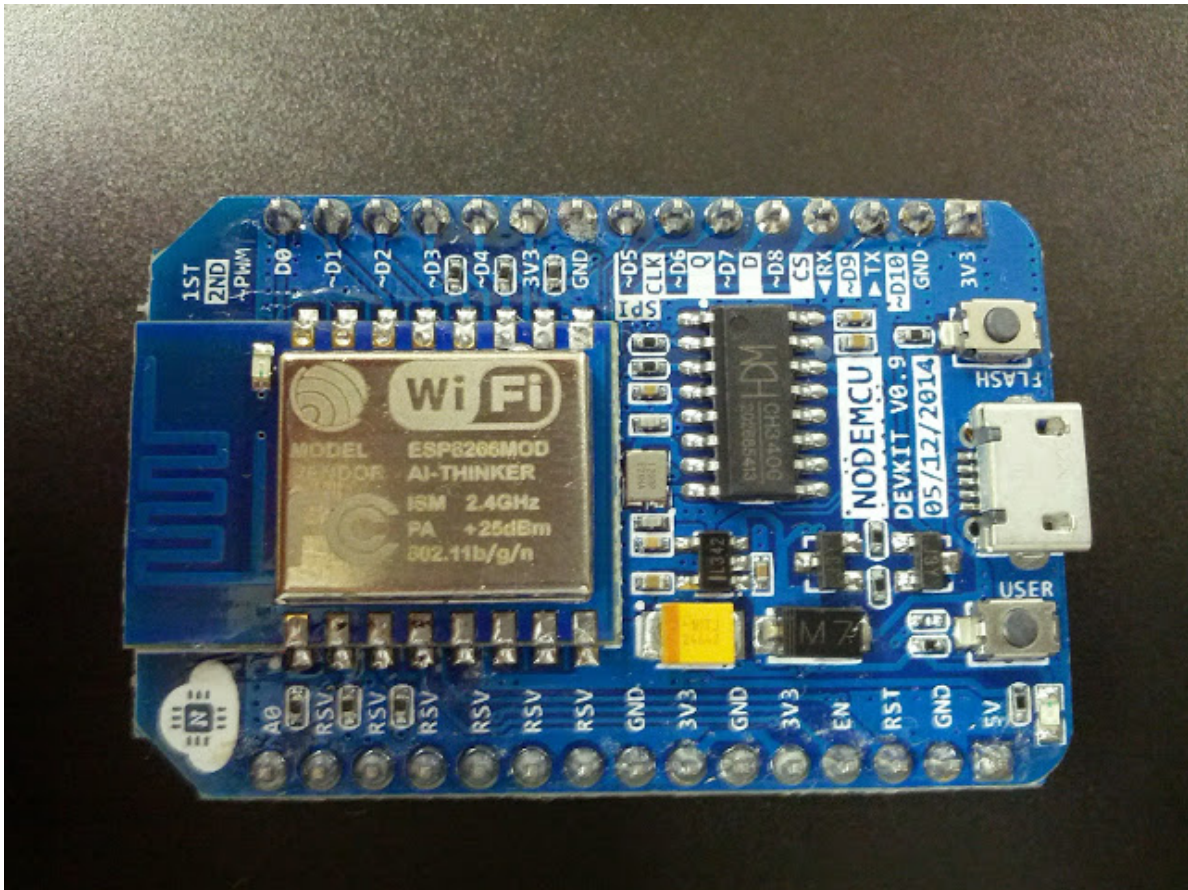


# มาเล่น NodeMCU devkit กัน

เอา NodeMCU devkit มาให้ลองเล่น บอร์ดตัวนี้เป็นบอร์ดที่รวมเอา ESP8266 (ESP-12) + USB to Serial + NodeMCU firmware เข้าไว้ด้วยกัน ทำให้การใช้งานง่ายมากขึ้น ไม่ต้องมีอุปกรณ์ต่อพ่วงเยอะ แถมมี GPIO เพิ่มเป็น 10 พอร์ต เพียงพอต่อการใช้งาน สำหรับ NodeMCU devkit ที่ได้มาเป็นบอร์ดเปล่าๆ ใช้ AT Command ในการสั่งงาน ชีวิตไม่อิสระเท่าไร แต่ทางผู้พัฒนา Board ก็มี NodeMCU Firmware ให้คุณสามารถเขียน Lua ใส่ลงไปได้ ทำให้การใช้งานง่ายมากขึ้น

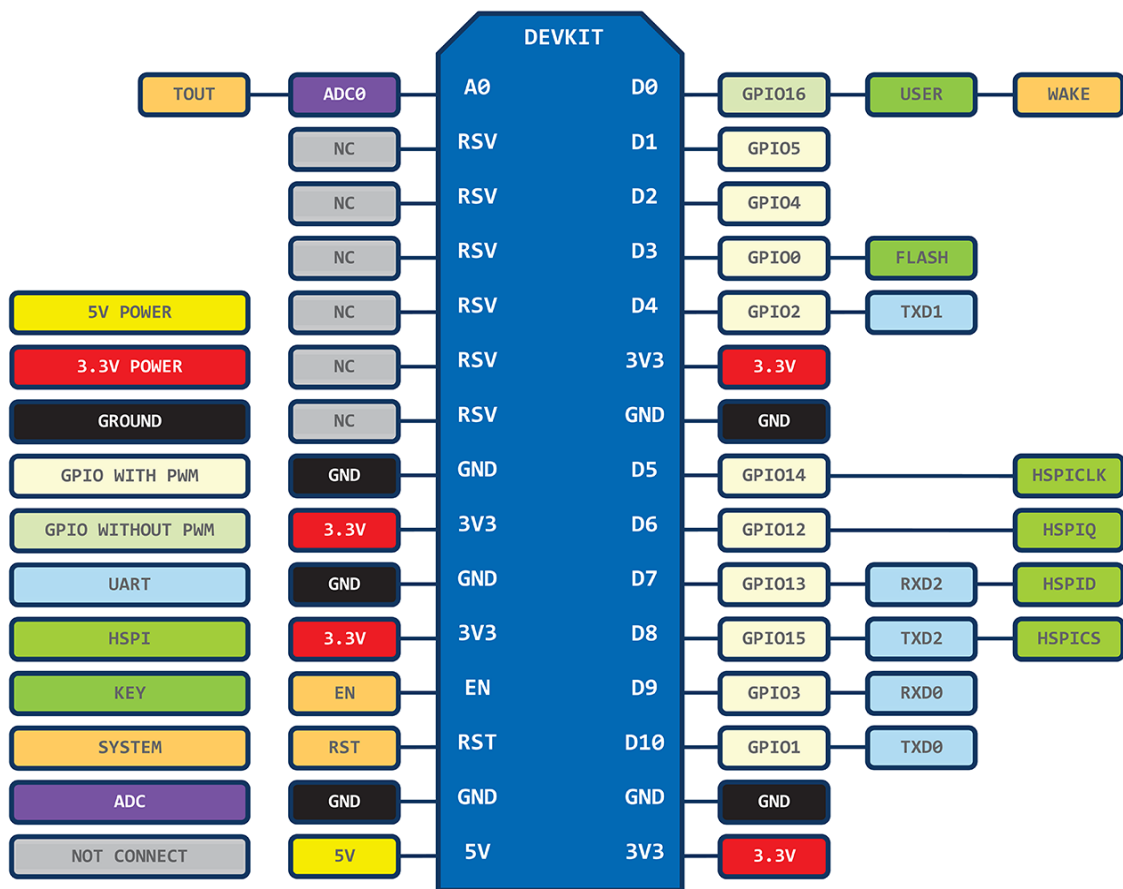












ท่านที่เคยเล่น ESP8622 โดยใช้ [Firmware ของ NodeCMU](#) ก็ปรับตัวได้ง่ายหน่อย เพราะวิธีการคล้ายๆ กัน ส่วนเครื่องมือที่ใช้มีหลายตัว ถ้าท่านที่ใช้ AT Command ก็ใช้ผ่าน Serial Monitor ของ Arduino Sketch ได้ เครื่องมือที่ต้องใช้ ได้แก่

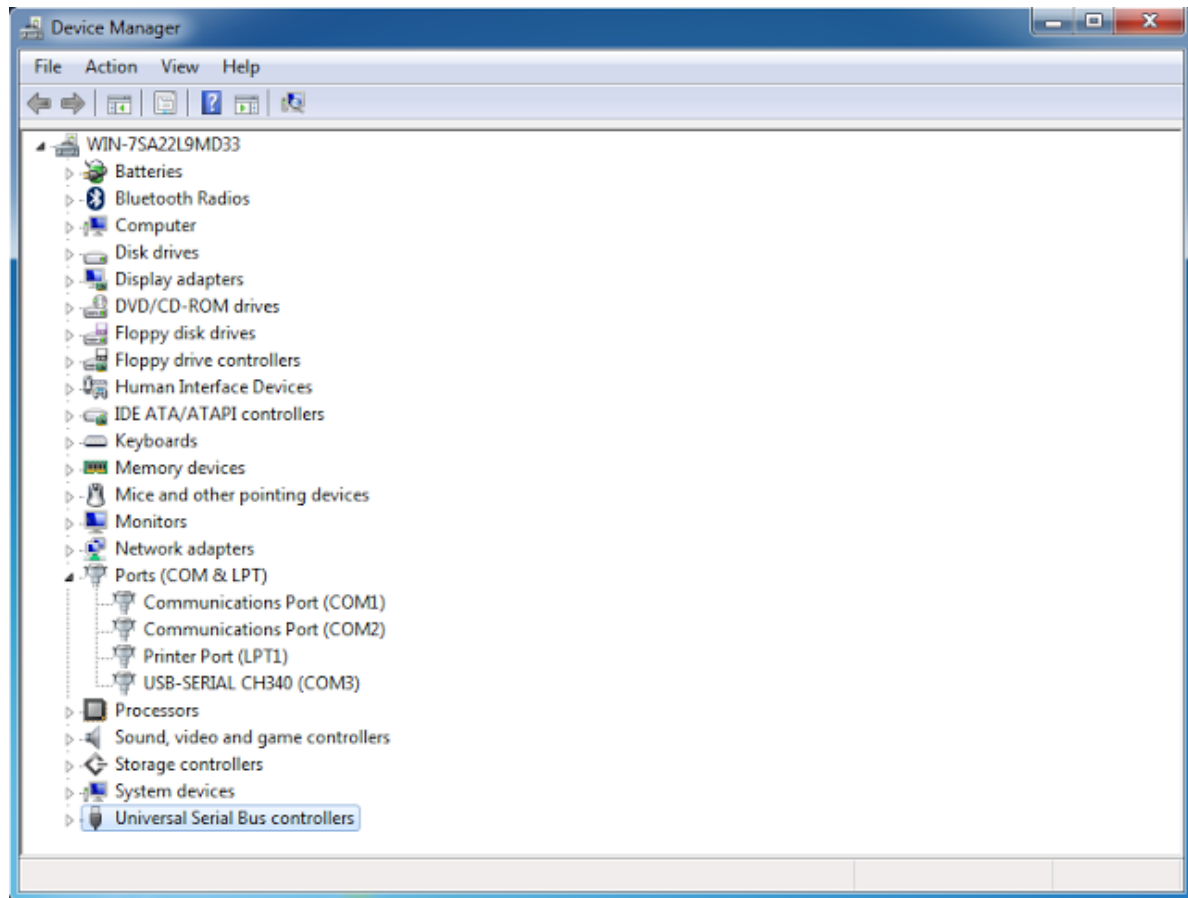
- [NodeMCU Studio](#)
- [Firmware Flasher](#)
- [CH340 Driver](#)
- [Lua Uploader](#)
- [.NET Framework](#)

หรือจะใช้เครื่องมืออื่นๆ ที่สามารถเชื่อมต่อแบบ Serial ได้ก็สามารถใช้ได้เช่นกัน

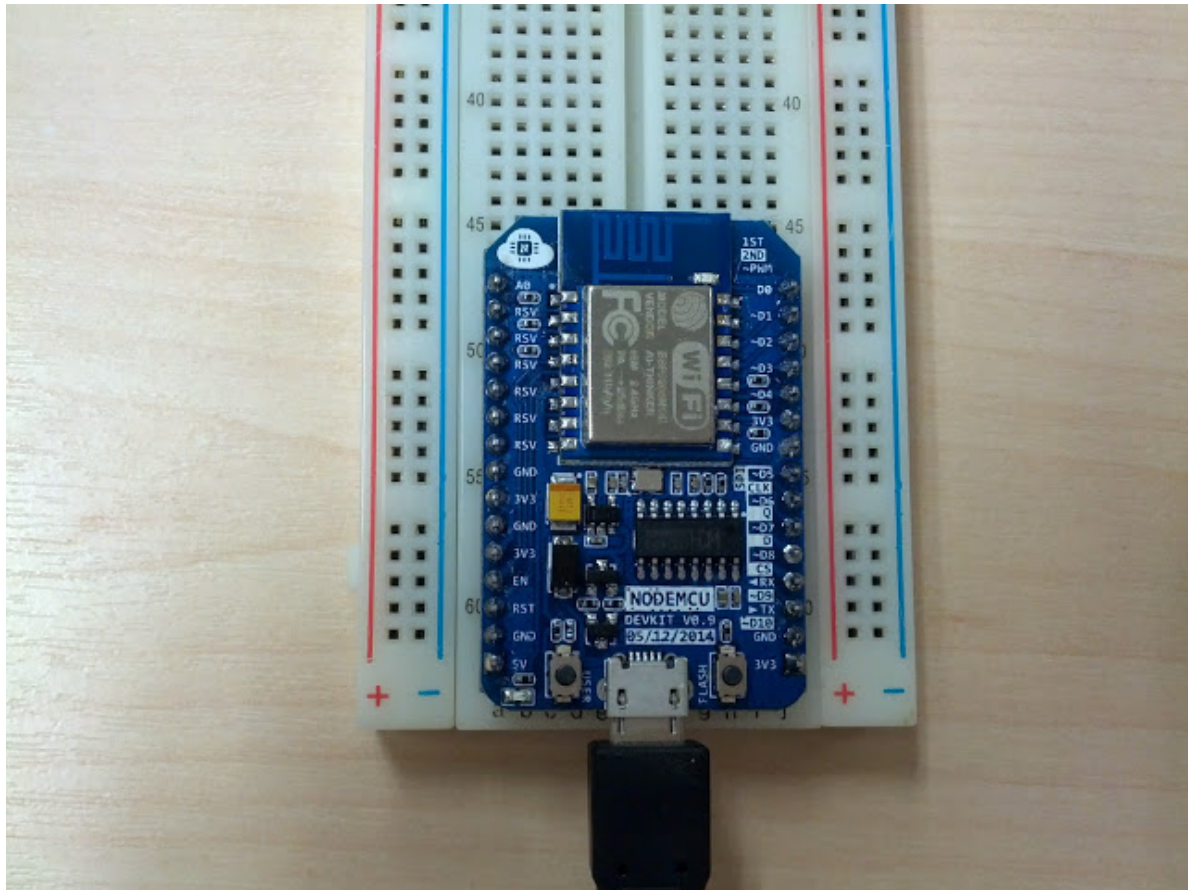
- TeraTerm
- CoolTerm
- SecureCRT
- Screen

สำหรับท่านที่ไม่อยากใช้ Lua ก็สามารถใช้ Arduino IDE ได้เช่นกัน ดูข้อมูลเกี่ยวกับการติดตั้ง board เพิ่มเติมที่ [เพิ่มบอร์ด ESP8266 ใน Arduino IDE](#) และท่านที่ใช้ NodeMCU Dev kit V2 ตัวสีดาอ่านวิธีการติดตั้งบอร์ดเพิ่มเติมที่บทความ [มาเล่น NodeMCU DevKit V2 กัน](#) สำหรับท่านที่ต้องการใช้ Lua อ่านต่อได้เลย

มาลองเล่นกัน อันดับแรกติดตั้ง driver USB to TTL กันก่อน



จากนั้นเสียบสาย USB เข้ากับ dev kit ของเราได้เลย



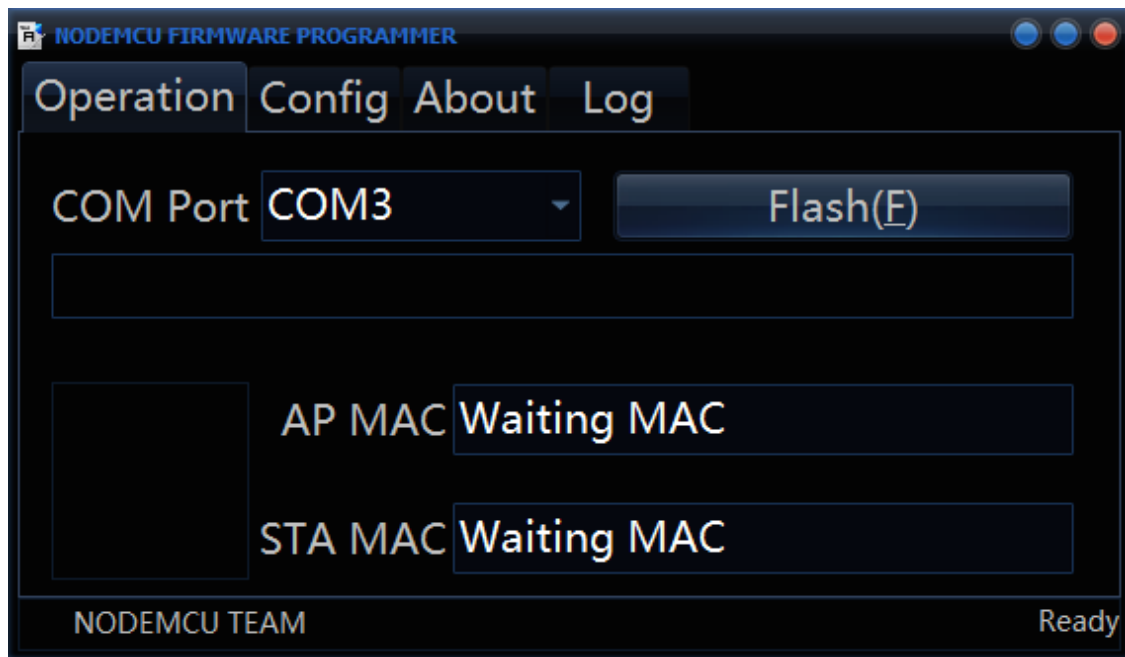
ใช้ terminal เชื่อมต่อไปยัง COM PORT ของผมได้ใช้ COM3 ตั้งค่า Board Rate เป็น 9600 เมื่อเชื่อมต่อแล้วลองใช้คำสั่ง AT เพื่อตรวจสอบว่า ESP8266 ใช้ firmware เดิมอยู่หรือไม่ หากพิมพ์ AT แล้วขึ้น ok แสดงว่าใช้ firmware ของ ESP8266 อยู่ หากขึ้น lua syntax error หรือแสดงรุ่นของ NodeMCU แสดงว่าใช้ firmware ของ NodeMCU อยู่ครับ

```

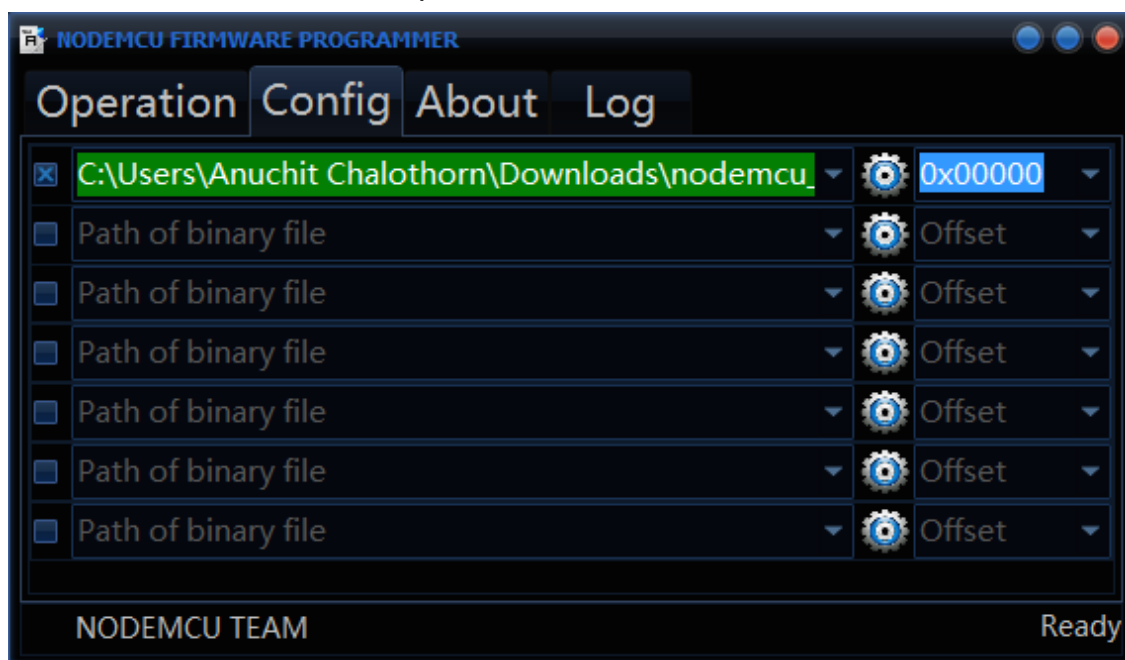
> IP:192.168.1.1 > >xdzn
NodeMcu 0.9.2 build 20141212 powered by Lua 5.1.4
lua: cannot open init.lua
>

```

สำหรับท่านใดที่ใช้คำสั่ง AT ไปแล้วขึ้น ok มา flash firmware ของ NodeMCU กันก่อน เปิด NodeMCU Flasher ขึ้นมา

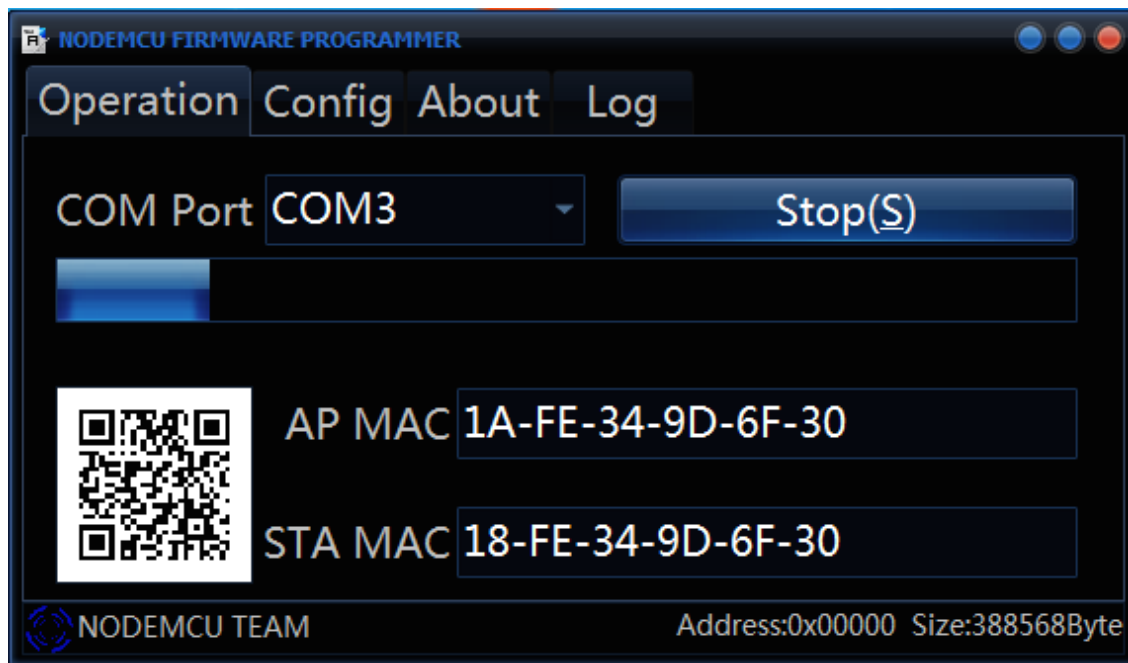


กด tab config กดปุ่มรูปเฟือง เลือก firmware รุ่นล่าสุดที่ดาวน์โหลดมา กำหนด Address เป็น 0x00000 ส่วน address อื่นๆ เลือก checkbox ข้างหน้าออก



กลับไปหน้า Operation แล้วกดปุ่ม Flash ได้เลย รอจน Flash เสร็จเรียบร้อย





ถอดสาย USB ออกแล้วเสียบเข้าไปใหม่ ลองใช้ terminal เชื่อมต่อเข้าไปใหม่ หน้าจอจะแสดงรุ่นของ NodeMCU แสดงว่าใช้ firmware ของ NodeMCU แล้ว

```

> IP:192.168.1.1 >xdzn
NodeMcu 0.9.2 build 20141212 powered by Lua 5.1.4
lua: cannot open init.lua
>

```

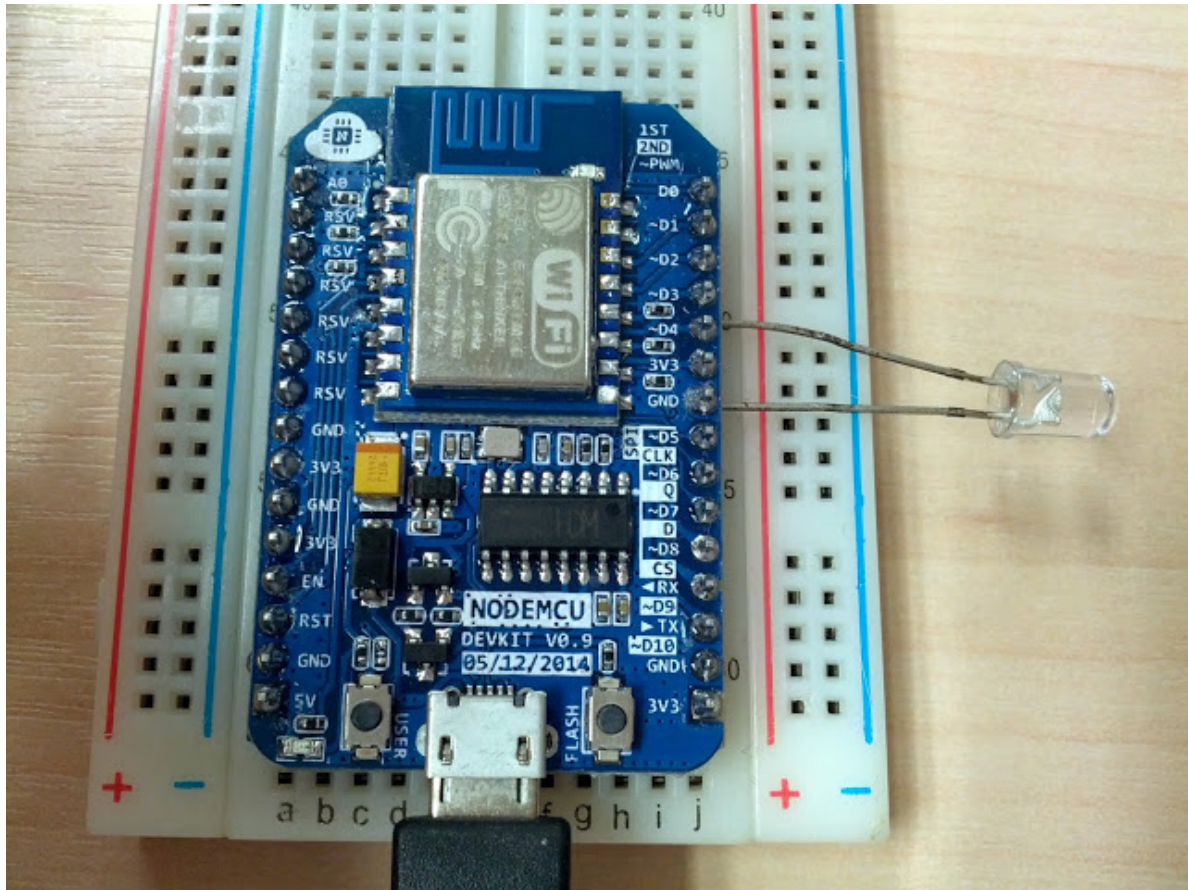
แสดงว่าตอนนี้ devkit ของเราใช้งาน lua ได้แล้ว สำหรับการทำงานกับ devkit สามารถใช้ Coolterm หรือเครื่องมือ terminal อื่นๆ ที่เชื่อมต่อผ่าน serial ได้ แต่แนะนำให้ใช้ Lua Uploader หรือ NodeMCU Studio จะสะดวกมากขึ้น มาลองโปรแกรมแรกกันก่อน HelloWorld ดังนี้

```

-- print Hello World text
print("HelloWorld")

```

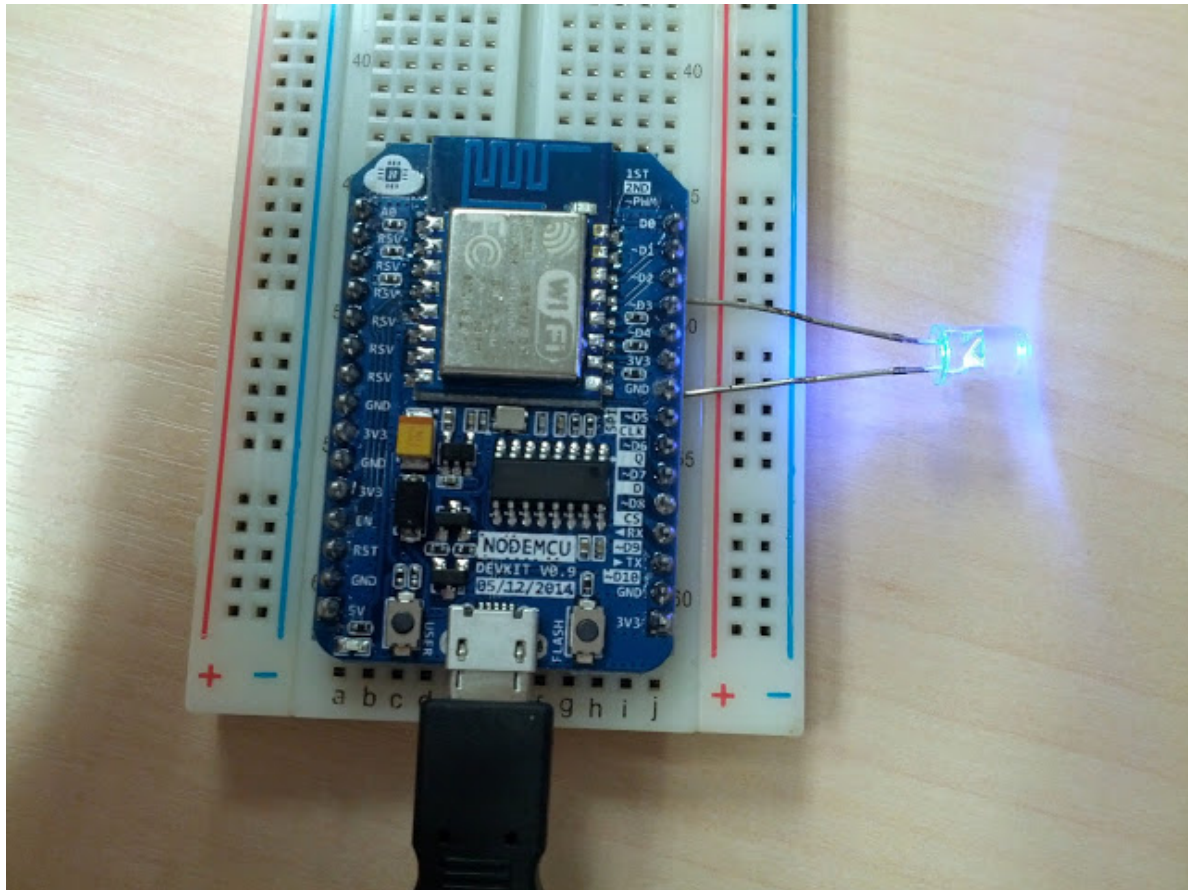
ภาษา Lua จะใช้ — แทน comment สำหรับท่านที่อยากศึกษาภาษา lua ก็ศึกษาเพิ่มเติมได้ที่ [lua.org](http://lua.org) มาลองตัวอย่างอื่นบ้าง สำหรับท่านที่ใช้ Arduino มาก่อนอาจคุ้นเคยกับการสั่งงานผ่าน digital pin, pwm ฯลฯ มาลองต่อวงจรกันดูครับ



จากภาพเราต่อ LED ที่ digital pin 4 หรือ D4 เขียนโค้ด lua ดังนี้

```
-- Blink using timer alarm --
timerId = 0
dly = 1000
-- use D4
ledPin = 4
-- set mode to output
gpio.mode(ledPin,gpio.OUTPUT)
ledState = 0
-- timer loop
tmr.alarm( timerId, dly, 1, function()
    ledState = 1 - ledState;
    -- write state to D4
    gpio.write(ledPin, ledState)
end)
```

ลอง run ดูหลอด LED กระพริบแล้ว :)



ทีนี้มาลองคุณสมบัติหลักของ ESP8266 บ้าง นั่นก็คือ Wifi นั่นเอง firmware มี module wifi มาให้ คุณสามารถสั่งให้เชื่อมต่อกับ Wifi Access Point ที่คุณต้องการได้ ดังนี้

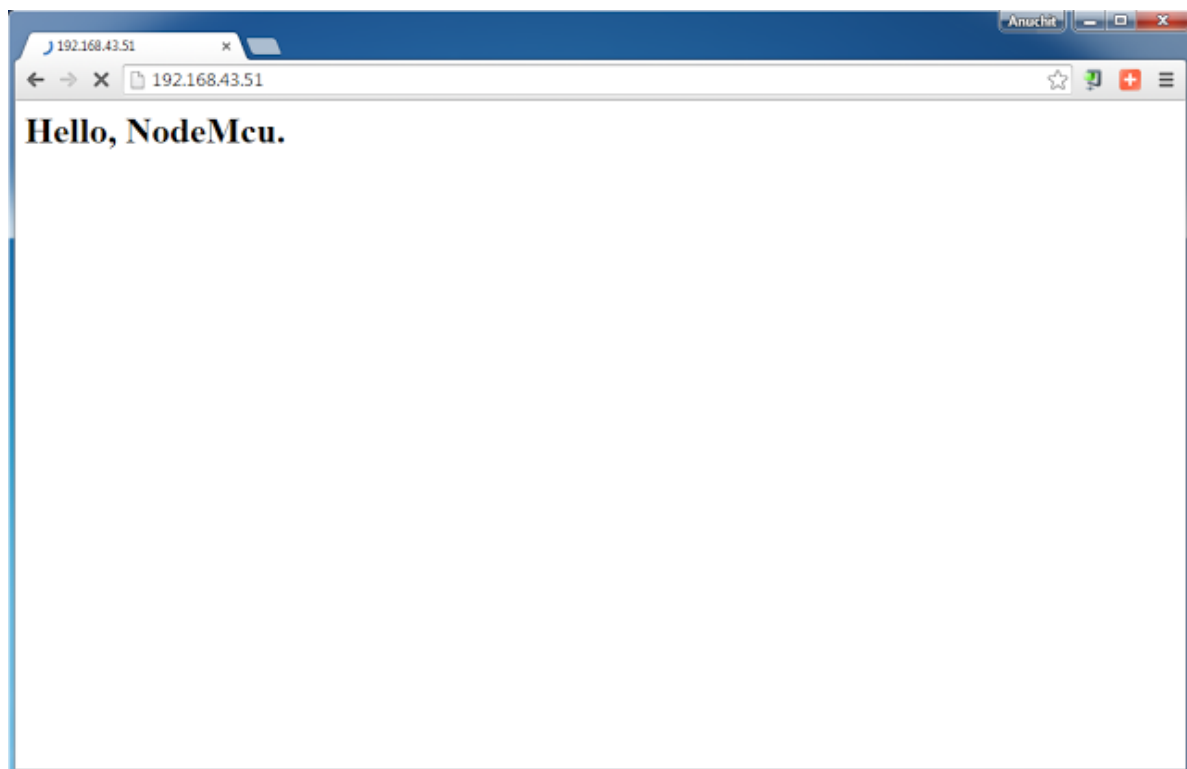
```
-- init.lua
print("set up wifi mode")
wifi.setmode(wifi.STATION)
--please config ssid and password according to settings of your wireless router.
wifi.sta.config("SSID","PASSWORD")
wifi.sta.connect()
cnt = 0
tmr.alarm(1, 3000, 1, function()
    if (wifi.sta.getip() == nil) and (cnt < 20) then
        print("IP unavaible, Waiting...")
        cnt = cnt + 1
    else
        tmr.stop(1)
        if (cnt < 20) then print("Config done, IP is "..wifi.sta.getip())
        --dofile("script.lua")
        else print("Wifi setup time more than 20s, Please verify
wifi.sta.config() function. Then re-download the file.")
        end
    end
end)
end)
```



แก้ไขค่า SSID และ PASSWORD เป็น Wifi Access Point ของคุณ ตัว devkit จะได้ ip address กลับมา มาลองคุณสมบัติแบบ IoT กันบ้าง เราจะลองสร้าง Web Server ง่ายๆ ก่อน

```
-- script.lua
-- a simple http server
print(wifi.sta.getip())
srv=net.createServer(net.TCP)
srv:listen(80,function(conn)
    conn:on("receive",function(conn,payload)
        print(payload)
        conn:send("Hello, NodeMcu.")
    end)
end)
```

เปิด browser ไปที่ devkit จะได้ผลลัพธ์ดังนี้



แก้ไขโค้ดแบบเร็วๆ อีกหน่อยให้สั่งงาน GPIO เพิ่มเติม

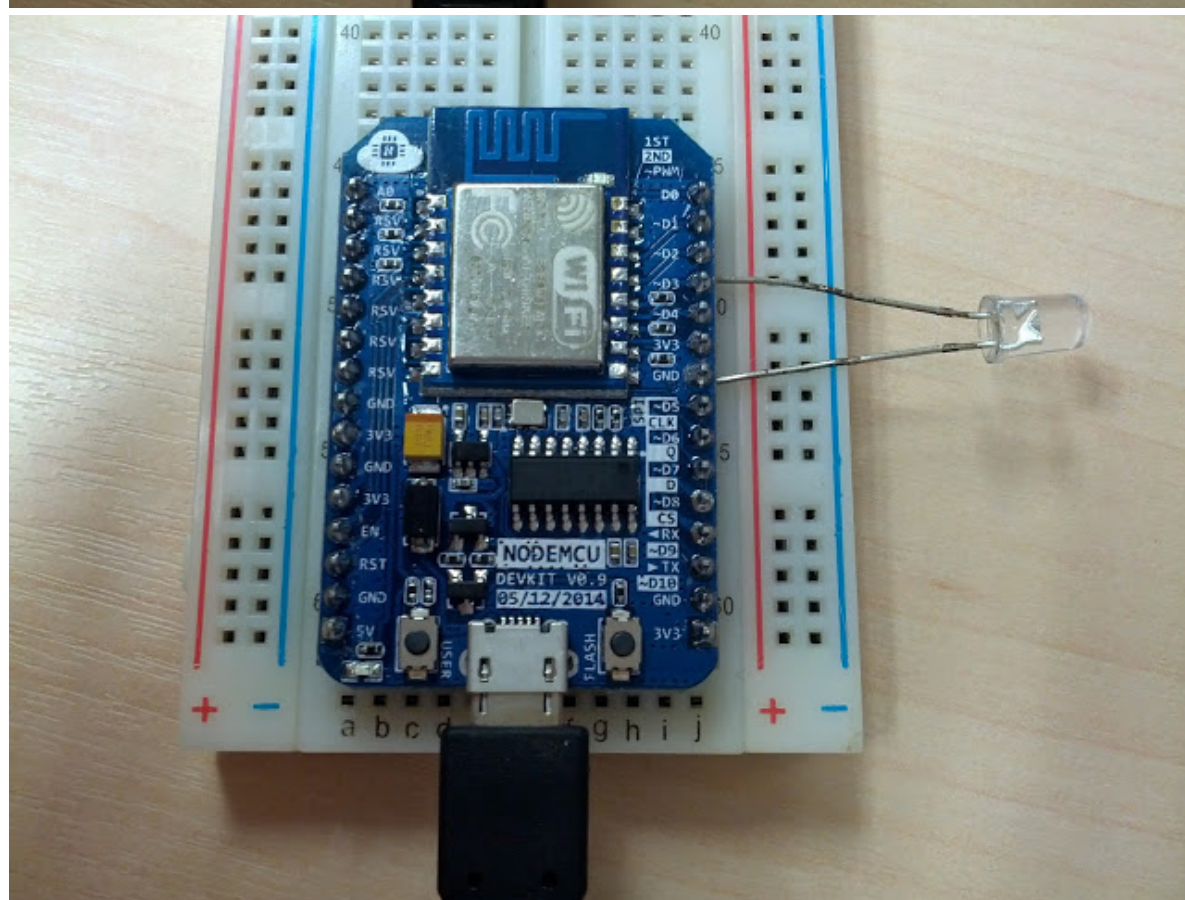
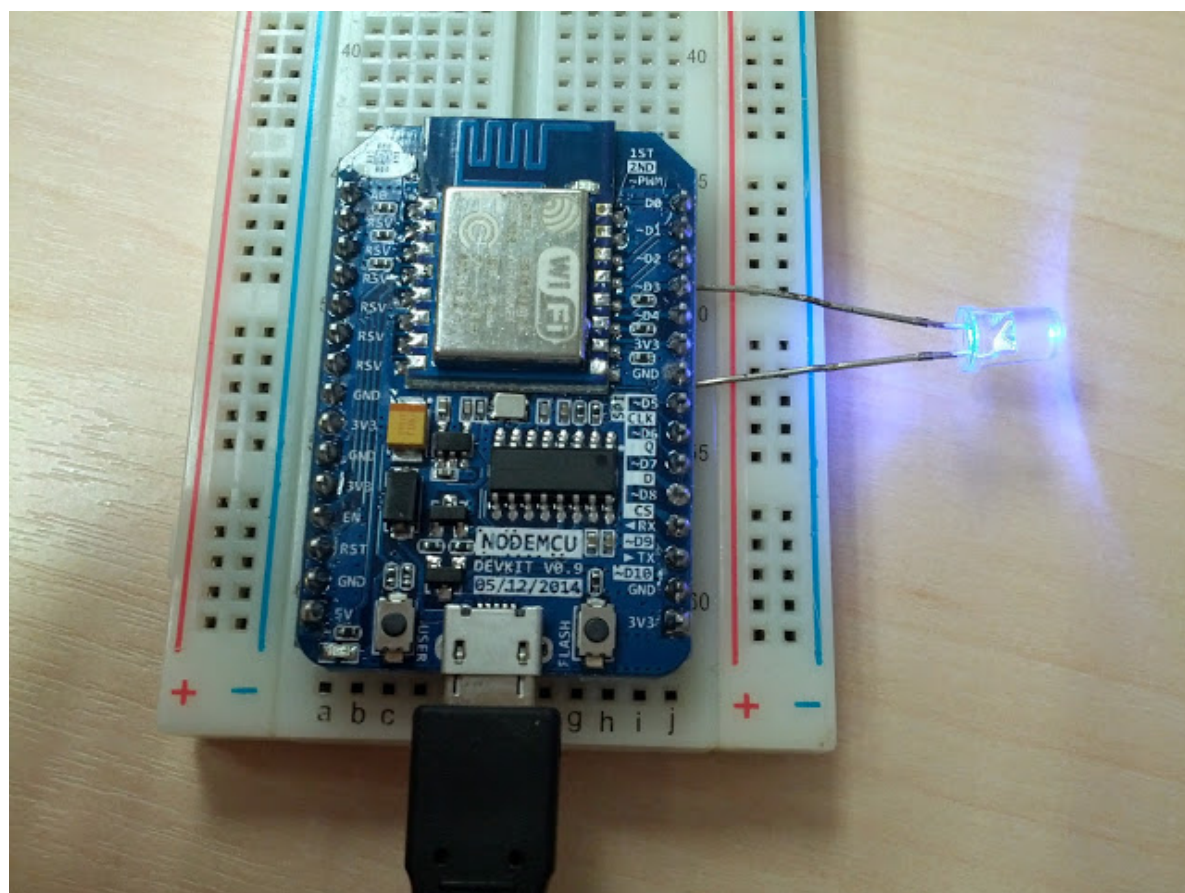
```
-- script.lua
-- a simple http server
srv=net.createServer(net.TCP)
srv:listen(80,function(conn)
    conn:on("receive",function(conn,payload)
        --print(payload)
        i,j=string.find(payload, "/d2/1", 1)
        pin=2
        gpio.mode(pin,gpio.OUTPUT)
        if (i==nil) then
            print("off")
        end
    end)
end)
```

```
        gpio.write(pin,gpio.LOW)
        conn:send("D2 : OFF")
    else
        print("on")
        gpio.write(pin,gpio.HIGH)
        conn:send("D2 : ON")
    end
end)
end)
```

เปิด browser สั่ง เปิด/ปิด LED ต่อกับ pin D2 โดยใช้ url ง่ายๆ ดังนี้

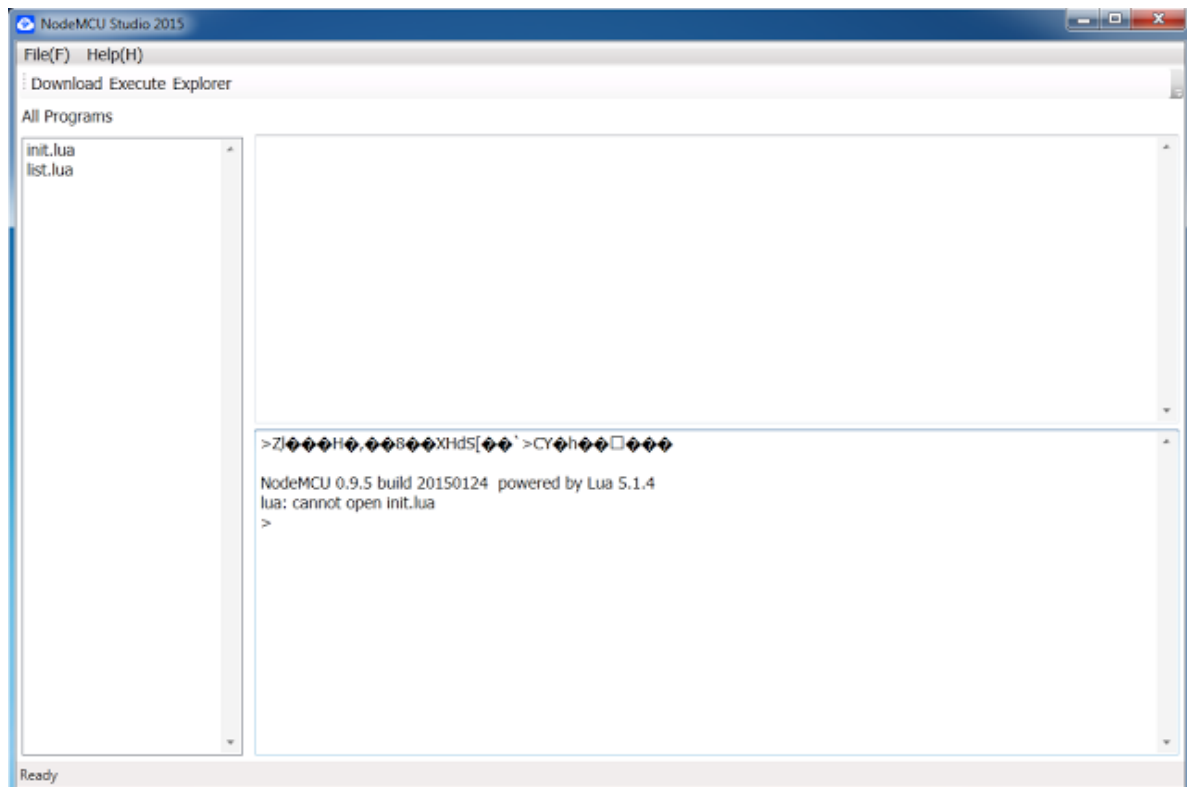
- เปิด <http://192.168.43.51/d2/1>
- ปิด <http://192.168.43.51/d2/0>

เปิด/ปิด LED ผ่าน Browser ได้ง่ายๆ ถ้าจะให้ดีก็เขียนเป็น REST API หรือใช้งานร่วมกับ MQTT Broker จะดีกว่า

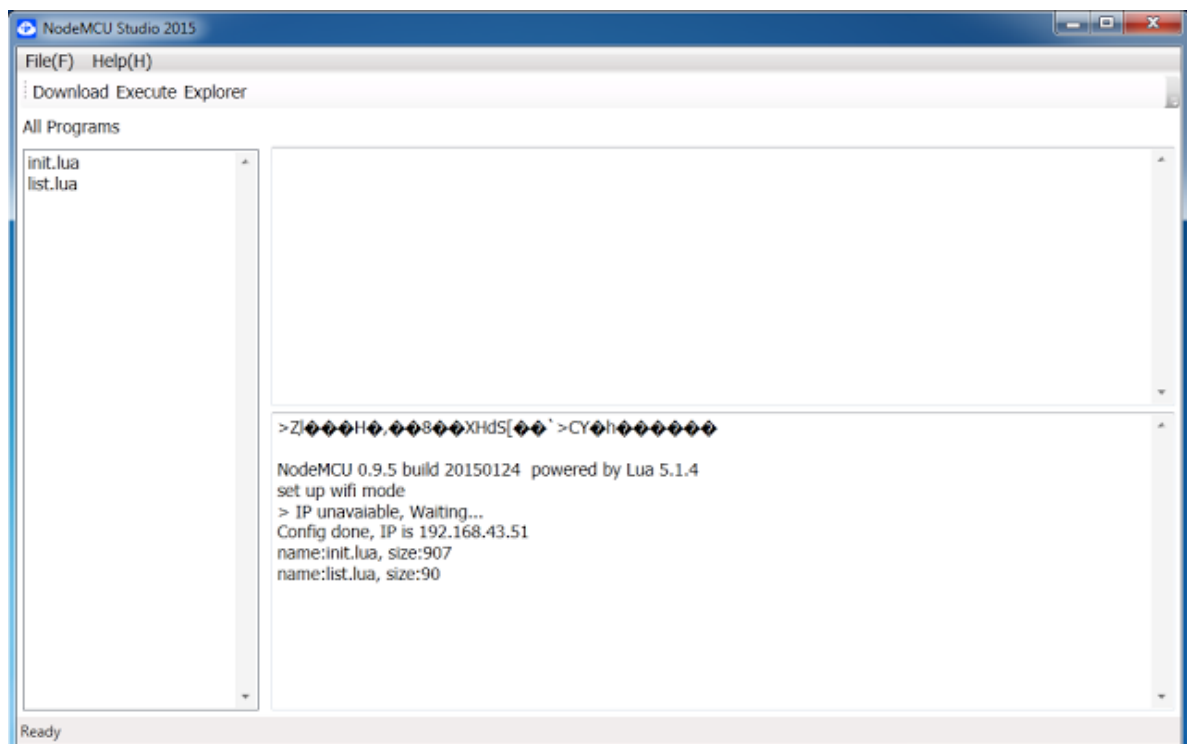




พอสนุกสนานครับ ก็กลับไปเรื่องหนึ่ง firmware ของ NodeMCU จะเรียกใช้งานไฟล์ init.lua เป็นไฟล์แรก หากไม่มี init.lua จะ error แบบนี้



แนะนำให้เขียนส่วนการเชื่อมต่อ Wifi ไว้ที่ไฟล์นี้ หากเชื่อมต่อสำเร็จค่อยสั่งให้ script ของเราไปเรียกอีกไฟล์หนึ่งขึ้นมาทำงาน ใช้ NodeMCU Studio จะสะดวกมากขึ้นครับ



สำหรับท่านที่สนใจอยากเล่น NodeMCU devkit ก็สามารถสอบถามจาก [Deaware](#) ได้ครับ  
สำหรับ API อื่นๆ อ่านเพิ่มเติมได้จากเอกสารของ [NodeMCU](#) ครับ