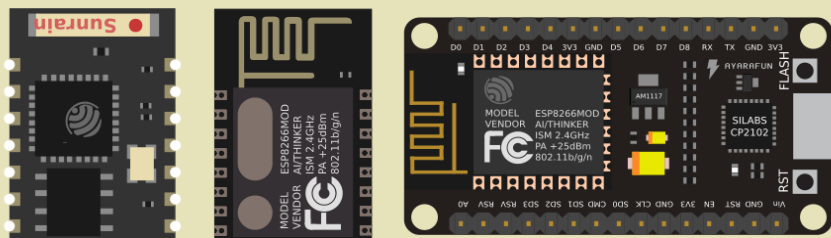


# รู้จักกับ Arduino ESP8266 (NodeMCU)



## ESP8266 Series ตอนที่ 1 - รู้จักกับบอร์ด ESP8266

เข้าสู่โลก internet of things (IoT) กันแล้ว โดยอุปกรณ์ที่มาแรง ในตอนนี้คงหนีไม่พลาด ESP8266 ซึ่งในตอนนี้ แดกกลูก แดกหลาน ออกมาหลายบอร์ด ทางเราก็เคยเอามาแนะนำไปแล้วในบทความเก่า [มาเล่น ESP8266 บน Arduino IDE กันเถอะ](#) ซึ่งบอร์ดที่เราใช้ เป็นบอร์ด NodeMCU Devkit ซึ่งในตอนนี้ ทาง NodeMCU จะออกมาอีกสองรุ่น คือ รุ่น 0.9 กับ รุ่น 1.0

ทางผู้พัฒนาดังใจจะออก NodeMCU ให้เป็น platform ที่ออกแบบทุกอย่างเป็น Node การทำงานย่อยๆ และ ใช้ภาษา Lua ในการเขียนโปรแกรม แต่ด้วย platform ที่สะดวกในการใช้งาน ทางกลุ่มนักพัฒนาของ ESP8266 ก็เลยนำ NodeMCU (ESP8266) มานับรวมในเป็นบอร์ดหนึ่งของ ARDUINO IDE (ESP8266) ด้วยเลย ได้จึงได้มีการพัฒนาต่อให้สามารถเขียนในภาษา C++ ซึ่งพลอยเองได้มาลองเริ่มเล่น หลังจากทีบอร์ด NodeMCU (ESP8266) นี้มีการพัฒนาบน ARDUINO IDE เรียบร้อยแล้ว หากเป็นผู้ที่นิยมเล่นไมโครคอนโทรลเลอร์อยู่ก่อนจะนิยมเล่นเป็นภาษา C/C++ ซึ่งภาษานี้สามารถไปได้กว้างเล่นได้หลายอย่างกว่า Lua



NodeMCU Devkit V1.0

**ความแตกต่างของลักษณะภาษาที่เขียนระหว่างภาษาLUA ใน LUA IDE และ ภาษา C/C++ บน Arduino IDE ในโปรแกรมเดียวกัน คือ สั่งให้ไฟกระพริบ (Blink)**

มาถึงตอนนี้ หลายคนมักมีคำถาม แล้ว Lua กับ C/C++ มันต่างกันอย่างไร Lua จะออกแบบให้เป็น interpreter มีคำสั่งการใช้งาน พอเพียงสำหรับใช้งานไมโครคอนโทรลเลอร์ แต่ด้วยมันเป็น Interpreter มันต้องใช้พื้นที่จำนวนมากใน flash rom ที่จะเอา firmware ตัวนี้ลงไปก่อน และ มันไม่ได้เป็น native ในบางเวลาจะทำงานช้ากว่า c/c++ แต่ข้อดีสุดๆ ของ LUA มันเป็น Interpreter จึงสามารถเขียนโปรแกรม จากเครื่องไหนก็ได้ ไม่ต้องพึ่ง compiler ต่อเข้ากับคอมได้ ก็เขียนโปรแกรมได้เลย

ส่วนใน C/C++ มีกลุ่มพัฒนา นำ SDK ของ ESP8266 มาพัฒนาต่อยอด ให้เข้ากับ platform ของ Arduino จึงทำให้ ESP8266 ใช้ภาษา C/C++ ได้นี่เอง

มันดูตัวอย่าง code กัน จะเห็นว่าความแตกต่างการใช้งานแล้วแต่ความชอบความถนัดของผู้ใช้

## And blink it

```
lighton=0
tmr.alarm(1,1000,1,function()
    if lighton==0 then
        lighton=1
        led(512,512,512)
    else
        lighton=0
        led(0,0,0)
    end
end)
```

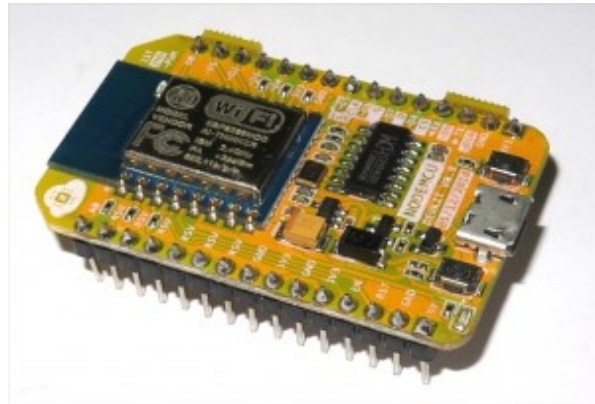
ภาษา LUA

ในภาษา Lua จะเขียนเป็นแบบ Event – Drive และ มองทุกอย่างเป็น Object ส่วนใน Arduino C/C++ หลายคน ที่ arduino มาอยู่แล้ว คงจะคุ้นเคยมากกว่า

ภาษา C/C++ บน Arduino IDE

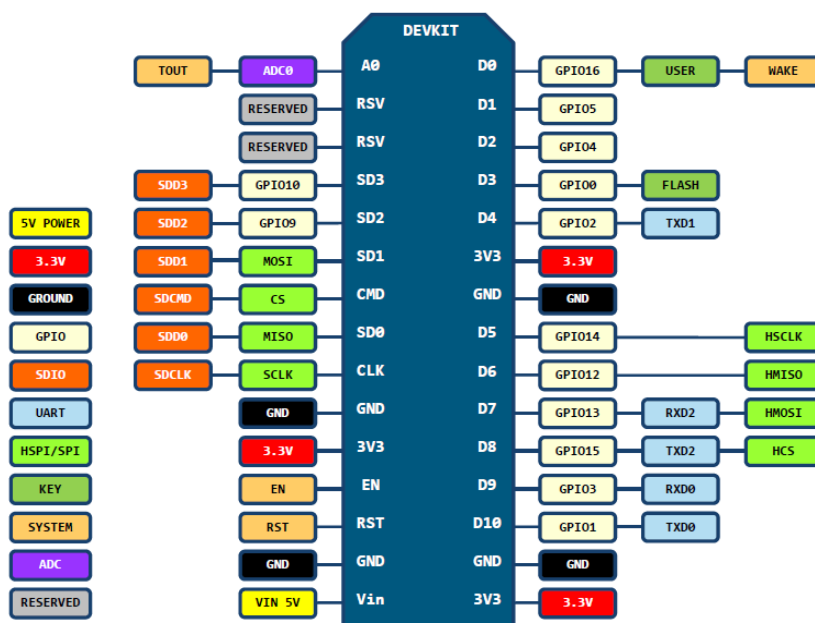
## ข้อดีของบอร์ด Arduino ESP8266

- เป็นแบบ Open Source Project มี Source code ให้ได้เรียนรู้อยู่บน Github ตามลิงค์ <https://github.com/esp8266/Arduino>
- สามารถกด upload sketch ได้ เชื่อมต่อบอร์ด USB กับคอมพิวเตอร์ใช้งานง่าย ขนาดของบอร์ดต่อลง Protoboard ได้
- ชิปภายใน ESP 8266 มี CPU ขนาด 32 bit แตกต่างจาก Arduino ที่เป็น CPU 8 bit
- ถึงแม้ขา I/O จะไม่มากเท่าของ Arduino แต่เราสามารถเขียนโปรแกรมลงบนขา GPIO ได้ทุกขาพอๆกัน เป็นข้อดีที่เพิ่มมาจากความต้องการใช้ WIFI เชื่อมต่อเมื่อต้องการเล่น Arduino ทำให้ต้องซื้อ Module wifi เพิ่ม นั่นคือ NodeMCU (ESP8266) มีต้นทุนต่ำกว่ามาก
- มีอุปกรณ์หลายอย่างที่ใช้งานที่แรงดัน +3.3 V เป็นส่วนใหญ่ ดังนั้นเราสามารถนำ NodeMCU (ESP8266) มาใช้เชื่อมต่อได้โดยตรง





## PIN DEFINITION



*D0(GPIO16) can only be used as gpio read/write, no interrupt supported, no pwm/i2c/ow supported.*

- NodeMCU V1.0 มีลักษณะยาวขึ้นแต่แคบลง สามารถเสียบลงบน protoboard ได้
- ชิฟแปลง USB2Serial ของ V0.9 เป็น CH340 แต่ของ V1.0 เป็น CP2102 ในเวอร์ชันแรก ไดรเวอร์จะมีปัญหา OS ค่อนข้างมาก แต่ใน CP2102 จะใช้บน MacOS และมีปัญหาน้อยกว่า
- ชิฟ WIFI ของ V0.9 เป็น ESP-12 แต่ใน V1.0 เป็น ESP-12E ทำให้มีจำนวนพินที่มากขึ้น และ พื้นที่ Flash มากขึ้น
  - ใช้ชิป Flash ความจุ 32Mbits (4MBytes)
- มีปุ่ม RST (รีเซ็ตการทำงาน) และ ปุ่ม Flash (สำหรับโปรแกรมเฟิร์มแวร์ใหม่)
- มีขา A0 รับอินพุตแรงดันแบบแอนะล็อกสำหรับวงจร ADC (ขนาด 10 บิต) อ่านค่า 0V ถึง 1V

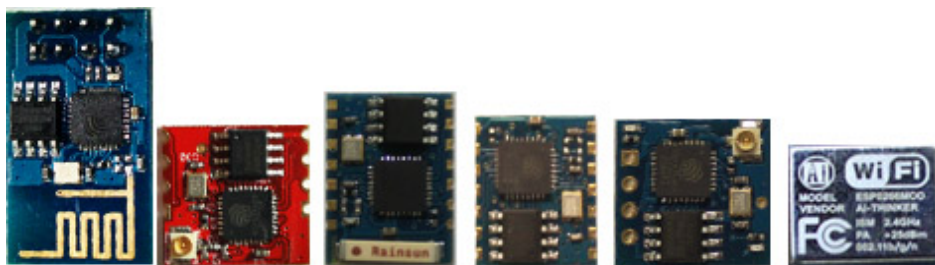
นอกจากนี้ ตัว Arduino ESP ยังใน กับ ESP8266 ทั่วไปๆ ได้ด้วย แต่เราจะไม่พูดถึงมาก มันต้องทำวงจร ใช้ในการ Flash เองเอง ซึ่ง ทางเราคิดว่า อยากให้สะดวก ได้ลองใช้กันเร็วๆ ดีกว่า ยุคนี้ Technology ไปกันไว้มากๆ



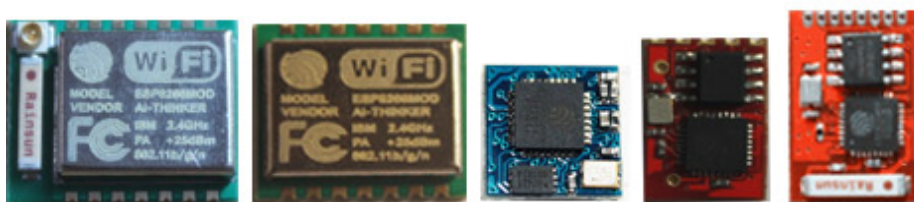
ชิป WiFi บน Node MCU ESP8266 Devkit 0.9



ชิป WiFi บน Node MCU ESP8266 Devkit V1



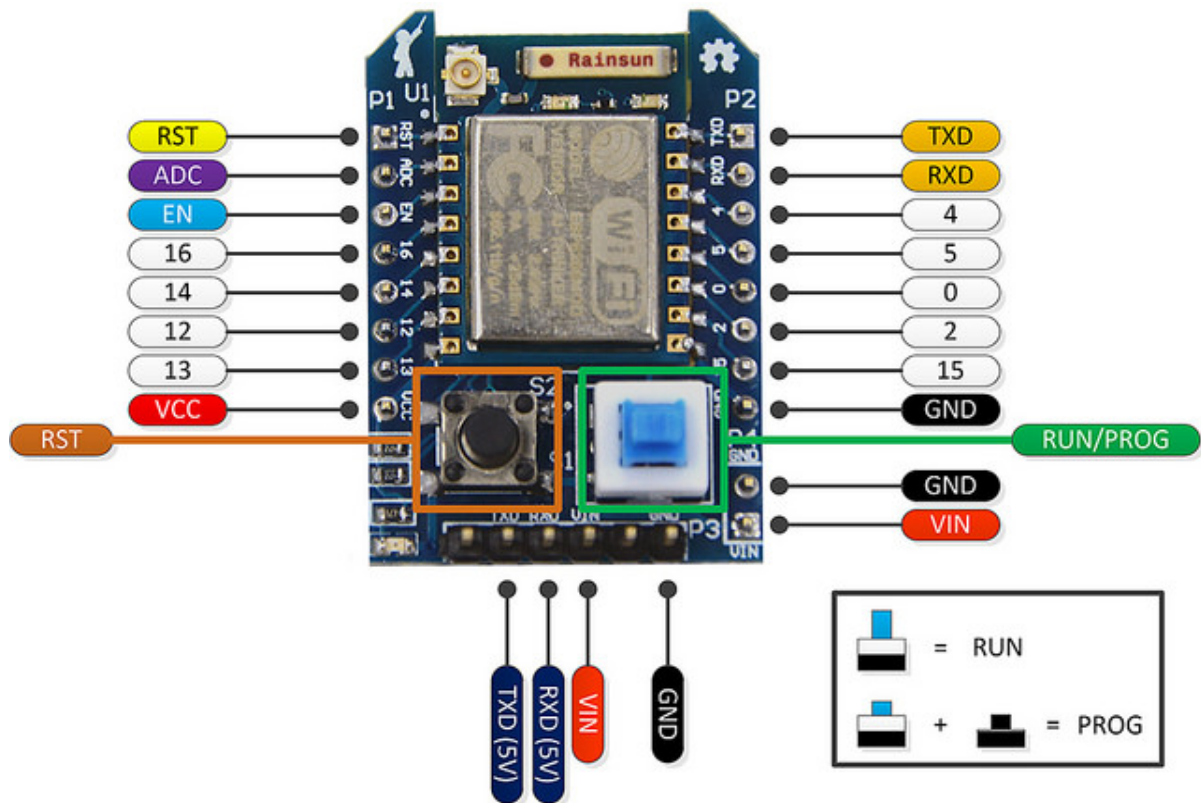
ESP-01 ESP-02 ESP-03 ESP-04 ESP-05 ESP-06



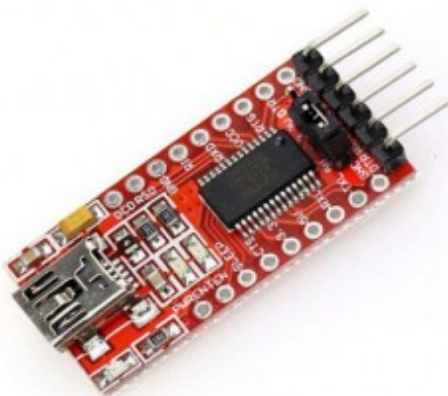
ESP-07 ESP-08 ESP-09 ESP-10 ESP-11

แต่นอกจาก NodeMCU devkit แล้วก็ยังสามารถนำตัวชิพ WiFi ESP-12E หรือ ESP-12 มาออกแบบวงจรทำบอร์ดเองได้ แต่ข้อเสียคือเราต้อง Flash firmware เข้าไปเพิ่ม อย่างบอร์ดนี้ที่เป็นของ dw.mini ESP8266 ใต้ใช้ FT232 เป็นตัว Flash แล้วเชื่อมต่อเข้าบอร์ดแบบ UART





บอร์ดทำเองของ dw.mini ESP8266 โดยใช้ชิพ WiFi แบบเดียวกับ Node mcu esp8266



ใช้ FT232 เป็นตัว Flash เข้าบอร์ด

จะเห็นว่าเรามีหลายตัวเลือกในการใช้งานมาก แต่เราจะเลือกใช้ Node MCU (ESP8266) เลย หรือก็คือ Arduino ESP8266 เพราะใช้งานง่าย สะดวก ประหยัด และที่สำคัญที่สุดเราไม่ต้อง Flash firmware อะไรเพิ่มก็ใช้งานได้เลย มาถึงตอนนี้ คงช่วยหลายคนสับสน Lua และ Arduino ESP8266 น้อยลงบ้างแล้วนะ โดยบทความต่อไป เราจะใช้ Arduino ESP8266 ทำโปรเจกต์