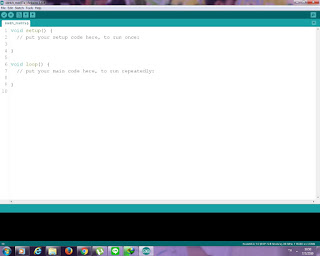
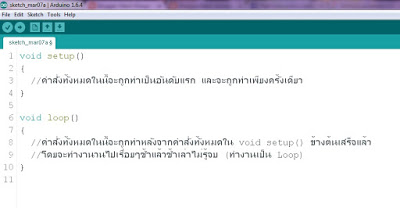
**พื้นฐานโปรแกรม Arduino**

[](https://4.bp.blogspot.com/-HNbIzdmcoL4/Vt15PNRmhcI/AAAAAAAAAo8/30vQqrnM00I/s1600/Untitled.jpg)

เมื่อเราเข้าสู่ Arduino ก็จะพบกับหน้าตาโปรแกรมแบบนี้นะครับ (เวอร์ชั่นอื่นอาจจะไม่เหมือนกันนะครับ) โดยที่ในส่วนสำหรับเขียนโปรแกรม จะมี **void (ในที่นี้ขอเรียกว่า "ชุดคำสั่ง")** อยู่ 2 void ครับ

เพื่อความดูง่าย ผมขอเคาะบรรทัดใหม่ให้เป็นแบบนี้นะครับ

[](https://2.bp.blogspot.com/-rHLil8Nugdk/Vt17EGCUEjI/AAAAAAAAApI/ihe3MY63vYY/s1600/Untitled.jpg)

**void setup()**ชุดคำสั่งนี้เป็นชุดคำสั่งแรกที่จะทำงานครับ ทุกโค้ดที่อยู่ภายในวงเล็บปีกกาของชุดคำสั่งนี้ จะถูกดำเนินการไปทีละบรรทัดครับ **ชุดคำสั่งนี้จะทำงานเพียงรอบเดียว**นะครับ

**void loop()** หลังจากทำชุดคำสั่งทั้งหมดของ void setup() เสร็จแล้ว ก็จะมาทำคำสั่งต่างๆ ภายในวงเล็บปีกกาของชุดคำสั่งนี้ครับ **โดยจะทำงานวนไปเรื่อยๆไม่รู้จบ (ทำงานเป็น loop)** ครับผม

ข้อความที่อยู่ด้านหลังเครื่องหมาย // (เครื่องหมาย Slash สองตัว) จะถูกละเว้นจากการประมวลผลนะครับ

(โปรแกรมจะไม่เอาไปคิดครับ เหมือนกับว่าไม่มีอยู่) ใช้สำหรับการสร้างคอมเมนต์ หรือคำอธิบายให้กับโค้ดของเราครับ อาจจะเพื่อความเข้าใจ หรือใช้ป้องกันความสับสนที่อาจเกิดขึ้นในโค้ดเราก็ได้ครับ

**STEP 2: คำสั่งต่างๆที่จะใช้งาน**

ทีนี้ เราก็จะมาทำความรู้จักกับ คำสั่งหลักๆ ที่เราจะใช้ในการเขียนโปรแกรมกันนะครับ

**คำสั่ง pinMode( หมายเลขขา , โหมด );**

เป็นคำสั่งที่ใช้ในกำหนดโหมดการทำงานของขาพอร์ต Digital ครับ ส่วนใหญ่ทำงานอยู่ใน void setup()

ว่าจะให้เป็นขาที่ใช้รับสัญญาณลอจิคเข้า (เป็น Input) หรือเป็นขาที่ส่งสัญญาณลอจิคออก (เป็น Output)

ซึ่งจะเป็นขาเข้าหรือขาออกนั้น ขึ้นอยู่กับการใช้งานร่วมกับอุปกรณ์ที่เรานำมาต่อกับขานั้นๆครับ

**ในบอร์ดตระกูล Arduino จะใช้การพิมพ์ขาแบบ 1 , 2 , 3 , 4 ....**

**แต่ใน NodeMCU V2 จะใช้การพิมพ์ขาแบบ D1 , D2 , D3 , D4 ....**

**ส่วนโหมด จะมีพิมพ์ INPUT กับ OUTPUT ครับ**

**ตัวอย่างการใช้งาน เช่น**

1.ผมต่อเซนเซอร์วัดอุณหภูมิเข้าที่ขา 4 ของ Arduino UNO ต้องใช้คำสั่ง

**pinMode( 4 , INPUT );**

เพราะขานี้ จะถูกใช้**เป็นขาที่รับสัญญาณ**ลอจิคที่เซนเซอร์ส่งมา เพื่อนำไปประมวลผลต่อครับ

2.ผมต่อรีเลย์เข้ากับขา 6 ของ NodeMCU V2 ต้องใช้คำสั่ง  
 **pinMode( D6 , OUTPUT );**

เพราะขานี้ จะถูกใช้**เป็นขาที่ใช้ส่งสัญญาณ**ลอจิค เพื่อไปสั่งการรีเลย์ครับผม

**จะเห็นได้ว่าไม่ยากเลยครับ แต่ให้เข้าใจว่าระหว่าง Arduino กับ NodeMCU V2**

**จะมีการพิมพ์หมายเลขขาแตกต่างกันแค่นั้นเองครับผม คาดว่าน่าจะพอเข้าใจการใช้**

**คำสั่ง pinMode นี้กันแล้วนะครับ ^\_^** ต่อไปก็...

**คำสั่ง digitalWrite( หมายเลขขา , สถานะ );**

เป็นคำสั่งที่ใช้ในการทำให้ขาพอร์ต Digital ที่เราสั่ง มีสถานะตามที่เราต้องการครับ **โดยมี 2 สถานะ**

**คือ High กับ Low** ซึ่ง **High คือ มีไฟ** ครับ หากสั่งให้ขาใดเป็น High ที่ขานั้นก็จะมีไฟสัญญาณลอจิค

ที่แรงดัน 5 โวลท์ (สำหรับ Arduino ส่วนใหญ่) หรือ 3.3v (สำหรับ NodeMCU V2 ของเรา)

ส่วนสถานะ **Low คือ ไม่มีไฟ**ครับ หากสั่งเป็น Low ที่ขานั้นจะมีแรงดันเป็น 0 โวลท์ (ไม่มีไฟ) ครับผม

**ตัวอย่างการใช้งาน** **เช่น** ผมจะสั่งให้รีเลย์ ซึ่งต่อกับ NodeMCU V2 อยู่ที่ขา 6 ทำงาน

หลังจากผมใช้คำสั่ง pinMode เพื่อกำหนดให้ขา D6 เป็นแบบ Output ใน void setup ไปแล้ว  
ก็จะใช้คำสั่งนี้ครับ

**digitalWrite( D6 , LOW );**

การสั่งด้วยคำสั่งนี้จะทำให้ที่ขา D6 มีแรงดันไฟที่ 0v (ไม่มีไฟ) ครับ ซึ่งจะปล่อยทำให้รีเลย์ทำงาน

(สถานะนี้ รีเลย์จะดึงขา C ไปต่อกับขา NO ครับผม)

และคำสั่งที่จะทำให้รีเลย์ (ตัวเดิม) หยุดทำงาน ก็คือคำสั่งนี้ครับ

**digitalWrite( D6 , HIGH );**

การสั่งด้วยคำสั่งนี้จะทำให้ที่ขา D6 มีแรงดันไฟที่ 3.3v (มีไฟ) ครับ ซึ่งจะไปทำให้รีเลย์หยุดทำงาน

(สถานะนี้ รีเลย์จะปล่อยให้ขา C กลับไปต่อกับขา NC ครับ)  
(อาจจะดูสวนทางกับความคิดนิดๆนะครับ ผมเองก็ไม่เข้าใจเขาเหมือนกันครับ 555+)

ต่อไปก็จะเป็นการหน่วงเวลาครับ โดยจะใช้คำสั่งนี้ครับ

**คำสั่ง delay( เวลาในหน่วยมิลลิวินาที );**

คำสั่งนี้เป็นคำสั่งที่ใช้ในการรอเวลาครับ โดยโปรแกรมจะรอให้เวลาผ่านไปโดยไม่ทำอะไร

ตามเวลาที่กำหนดครับ**โดยมีหน่วยของเวลาเป็น มิลลิวินาที (ms)** นะครับ

**เช่น 1 วินาทีก็จะเท่ากับ 1000 มิลลิวินาทีครับ 0.5 วิ ก็จะเท่ากับ 500 ms ครับผม**

**ตัวอย่างการใช้คำสั่ง**

จากคำสั่งข้างบน ผมต้องการให้รีเลย์ทำงานเป็นเวลา 5 วินาที แล้วหยุดการทำงาน

โดยใช้ร่วมกับคำสั่ง digitalWrite ข้างบนนะครับ จะเป็นอย่างงี้ครับ

**digitalWrite( D6 , LOW );**

**delay(5000);**

**digitalWrite( D6 , HIGH );**

จากการใช้คำสั่ง 3 บรรทัดข้างบนนี้ ผลที่ได้คือ รีเลย์จะเริ่มการทำงาน

แล้วจะรอเวลา 5 วินาที แล้วรีเลย์ก็จะหยุดทำงานครับ

**สำหรับลำโพงเปียโซของเรา จะมีเสียงดังเมื่อสัญญาณลอจิคเป็น Low**  
**และจะหยุดส่งเสียงเมื่อสัญญาณลอจิคเป็น High นะครับ (ใช้สถานะเหมือนของรีเลย์ครับ)**

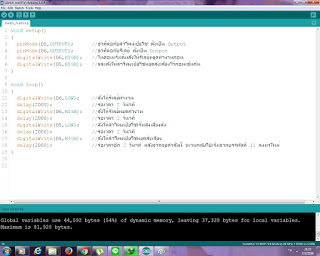
เพียงแค่ 3 คำสั่งนี้ก็เพียงพอให้อุปกรณ์ของเราทำงานได้แล้วครับ คำสั่งที่เหลือในโค้ดของ **NETPIE**

เราจะไม่เข้าไปยุ่งวุ่นวายอะไรกับเขานะครับ 555+ เราเพียงแต่จะสร้างชุดคำสั่งของเราไปผสม

กับโค้ดของเขาเฉยๆ ครับ ~~สำหรับคำสั่งอื่นๆนอกเหนือจากนี้ก็ให้ไปศึกษากันเองนะครับ (อ้าว = =" 55+)~~

สำหรับใครทีเซียนแล้วบทความในช่วงนี้ก็คงน่าเบื่อนิดๆนะครับ เพราะมันก็เบสิคอีกตามเคยครับผม TT

ทีนี้ก็จะมาที่ การสร้างโค้ดทดสอบกันนะครับ โดยโค้ดทดสอบของผมจะสั่งให้รีเลย์ทำงาน  
เป็นเวลา 3 วินาที และหยุดทำงานไป 3 วินาที หลังจากนั้นก็ให้ลำโพงเปียโซดังขึ้น 3 วินาที  
แล้วก็ดับไปเป็นเวลา 3 วินาที แล้วก็จะวนคำสั่งกลับไปเริ่มที่รีเลย์ทำงาน 3 วิใหม่อีกครั้ง  
วนไปวนมาเรื่อยๆครับ สำหรับโค้ดก็ตามนี้เลยนะครับ....

[](https://2.bp.blogspot.com/-cjJj378bfwE/Vt2NfQrsUXI/AAAAAAAAAqI/V3UcQRRm1os/s1600/Untitled.jpg)

อันนี้สำหรับคนที่ขี้เกียจพิมพ์นะครับ

**ก๊อบไปวางใน Arduino แล้วก็ เสียบสาย + กดอัพโหลด**

**แล้วดูผลลัพธ์ของโปรแกรมได้เลยครับผม ^\_^**