

## pwm code คืออะไร

PWM ย่อมาจาก Pulse Width Modulation (การจัดการความกว้างของช่วงช่วงสัญญาณ) ซึ่งเป็นเทคนิคในการควบคุมความสูงหรือความกว้างของสัญญาณไฟฟ้าเพื่อควบคุมความสัมพันธ์ระหว่างเวลาที่สัญญาณเปิดและปิดในระยะเวลาที่แน่นอน นี่เป็นวิธีที่พบบ่อยในการควบคุมความเร็วของมอเตอร์ที่ใช้ในหลายแอปพลิเคชัน เช่น รถยนต์, โดรน, เครื่องปรับอากาศ, และอุปกรณ์อื่น ๆ ที่ต้องการการควบคุมความเร็วหรือความสัมพันธ์ของเครื่องกลหรืออุปกรณ์อิเล็กทรอนิกส์อื่น ๆ ในทางปฏิบัติ

การใช้ PWM, คุณสามารถควบคุมปริมาณพลังงานที่ถูกส่งออกโดยเปลี่ยนความยาวของช่วงเวลาที่สัญญาณเปิด (ส่งพัลส์) และปิด (ไม่ส่งพัลส์) โดยอัตโนมัติ นี่สามารถช่วยให้คุณควบคุมความสัมพันธ์ระหว่างสัญญาณอินพุตและเอาต์พุตเพื่อควบคุมความสว่างหรือความเร็วของอุปกรณ์ที่มีการเชื่อมต่อกับระบบ PWM ในการเขียนโปรแกรมหรือการควบคุม PWM, คุณจะกำหนดความยาวของช่วงเวลาที่สัญญาณเปิดและปิด ซึ่งเรียกว่า "duty cycle" โดยใช้ค่าที่ระบุเป็นเปอร์เซ็นต์ ตัวอย่างเช่น หากคุณต้องการ PWM 50% duty cycle, คุณจะสั่งให้สัญญาณเปิด 50% ของเวลาและปิด 50% ของเวลาในรอบคลื่น นี่ช่วยให้คุณควบคุมความสว่างหรือความเร็วของอุปกรณ์ต่าง ๆ ได้อย่างยืดหยุ่นและมีประสิทธิภาพมากยิ่งขึ้น.

## การทำงานของโค้ด

```
Const int LEDpwm = 2;

void setup() {
  pinMode(LEDpwm, OUTPUT);
  Serial.begin(115200);
  analogWriteFreq(500);
}

void loop() {
  for(int pwmValue=0; pwmValue <= 1023; pwmValue++){
    analogWrite(LEDpwm, pwmValue);
    delay(1);
  }
  delay(10);
  for(int pwmValue=1023; pwmValue >= 0; pwmValue--){
    analogWrite(LEDpwm, pwmValue);
    delay(1);
  }
}
```

โค้ดนี้ใช้ Arduino IDE ในการควบคุมความสว่างของ LED โดยใช้ PWM (Pulse Width Modulation) บนขา  
ดังที่กำหนดในตัวแปร LEDpwm (ขาที่ต่อกับ LED).

## นี่คือวิธีที่โค้ดทำงาน:

โค้ดนี้ใช้ Arduino IDE ในการควบคุมความสว่างของ LED โดยใช้ PWM (Pulse Width Modulation) บนขา  
ดังที่กำหนดในตัวแปร `LEDpwm` (ขาที่ต่อกับ LED).

## นี่คือวิธีที่โค้ดทำงาน:

1. ``const int LEDpwm = 2;``: กำหนดขาที่ใช้สำหรับเชื่อมต่อ LED และตั้งชื่อของขาเป็น ``LEDpwm``.
2. ``void setup()`:` ฟังก์ชัน ``setup`` ใช้ในการตั้งค่าเริ่มต้น เมื่อ Arduino ถูกเริ่มต้นขึ้น:
  - ``pinMode(LEDpwm, OUTPUT);``: กำหนดขา ``LEDpwm`` เป็นแบบ OUTPUT เพื่อใช้ในการส่งสัญญาณออกไปที่ LED.
  - ``Serial.begin(115200);``: เปิดการสื่อสารระหว่าง Arduino และคอมพิวเตอร์ผ่านทาง Serial Communication ที่อัตราการส่งข้อมูลเท่ากับ 115200 bps (bits per second).
  - ``analogWriteFreq(500);``: กำหนดความถี่การ PWM เป็น 500 Hz (เฮิรตซ์).
3. ``void loop()`:` ฟังก์ชัน ``loop`` ทำงานซ้ำๆ ตลอดเวลา:
  - ``for(int pwmValue=0; pwmValue <= 1023; pwmValue++)``: เริ่มจากค่า ``pwmValue`` เท่ากับ 0 และเพิ่มค่า ``pwmValue`` ในการวนลูปจนถึง 1023 (ค่าสูงสุดของ PWM).
  - ``analogWrite(LEDpwm, pwmValue);``: กำหนดความสว่างของ LED โดยใช้ค่า ``pwmValue`` ที่ได้ในรอบปัจจุบันของลูป.
  - ``delay(1);``: รอเป็นเวลา 1 มิลลิวินาทีเพื่อให้ความสว่างเปลี่ยนแปลงอย่างช้าๆ.เมื่อค่า ``pwmValue`` เพิ่มขึ้นจนถึง 1023 แล้ว, จะมีการรอครั้งหนึ่ง (ในที่นี้คือ ``delay(10)``) แล้วค่า ``pwmValue`` จะลดลงจาก 1023 ไปสู่ 0 ด้วยความช้าเช่นเดียวกัน เมื่อค่า ``pwmValue`` เป็น 0 แล้ว จะมีการรออีกครั้ง (ในที่นี้คือ ``delay(10)``) และการวนลูปจะเริ่มใหม่.  
ผลลัพธ์ของโค้ดนี้คือ LED จะเริ่มที่ความสว่างต่ำ (0) และความสว่างจะเพิ่มขึ้นเป็นระยะเวลาและความสว่างสูงสุดที่ความสว่างเป็น 1023 แล้วจะลดลงอย่างช้าๆ จนถึง 0 และทำซ้ำอย่างต่อเนื่อง.