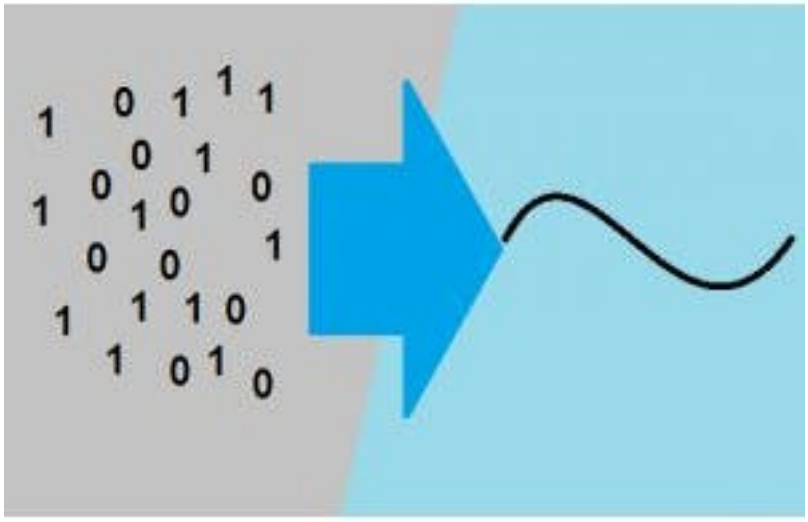
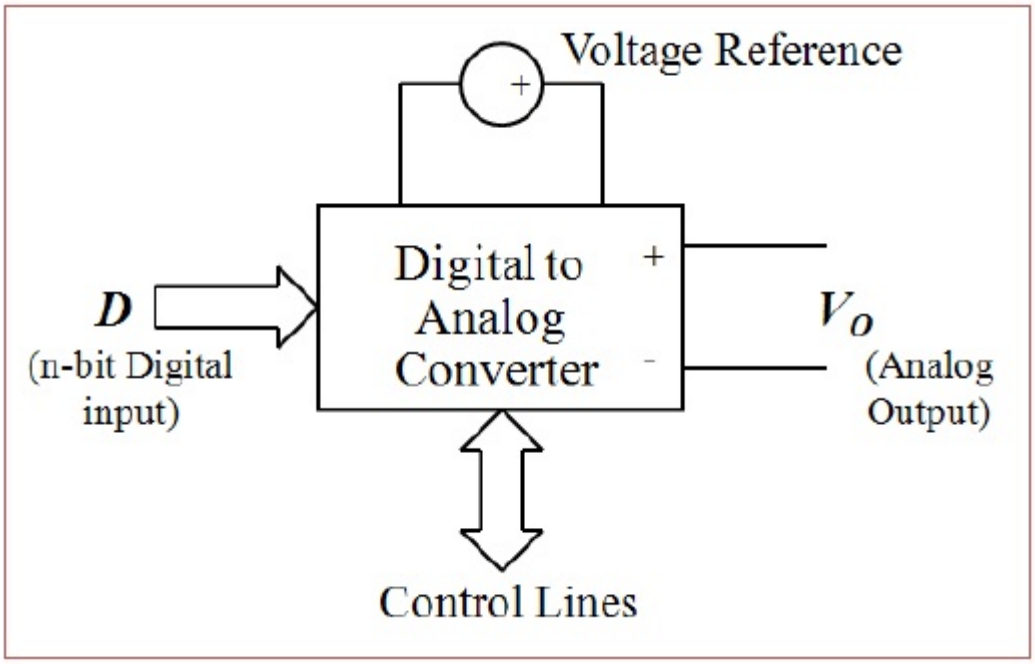
ใบความรู้ที่ 7 การใช้ thingcontrol board กับเอาท์พุตแบบอนาลอก

ไมโครคอนโทรเลอร์ และ คอมพิวเตอร์ เป็นอุปกรณ์ที่อยู่ในกลุ่มดิจิตอลอิเล็คทรอนิคส์ จะการทำงานด้วยข้อมูล “0” และ “1” เท่านั้น ซึ่งจะทำงานในลักษณะ เปิด – ปิด ได้ เช่น เปิด – ปิดหลอดไฟ, มอเตอร์ไฟฟ้า แต่ไม่สามารถทำงานในลักษณะการหรี่หลอดไฟได้ , ทำเสียงในโทนต่าง ๆ ได้ และการควบคุมความเร็วของมอเตอร์กระแสตรง จึงมีความจำเป็นขบวนการทำงานที่แปลงสัญญาณดิจิตอล เป็นสัญญาณอนาล๊อกเสียก่อนที่เรียกว่า Digital to Analog conversion (DAC)



ตัวแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอก (Digital-to-Analog Converter)

ตัวแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอก จะเป็นวงจรอิเล็คทรอนิคส์ที่ทำการรับค่าอินพุตแบบดิจิตอลในรูปเลขฐานสอง และทำการแปลงเป็นสัญญาณอนาลอก ในการแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอก เราจำเป็นที่ต้องรู้แรงดันอ้างอิง (Voltage Reference)



ความสัมพันธ์ของสัญญาณดิจิตอลอินพุตกับสัญญาณอนาลอกเอาท์พุตมีค่าดังนี้

$V_o = D/2^n \times V_r$

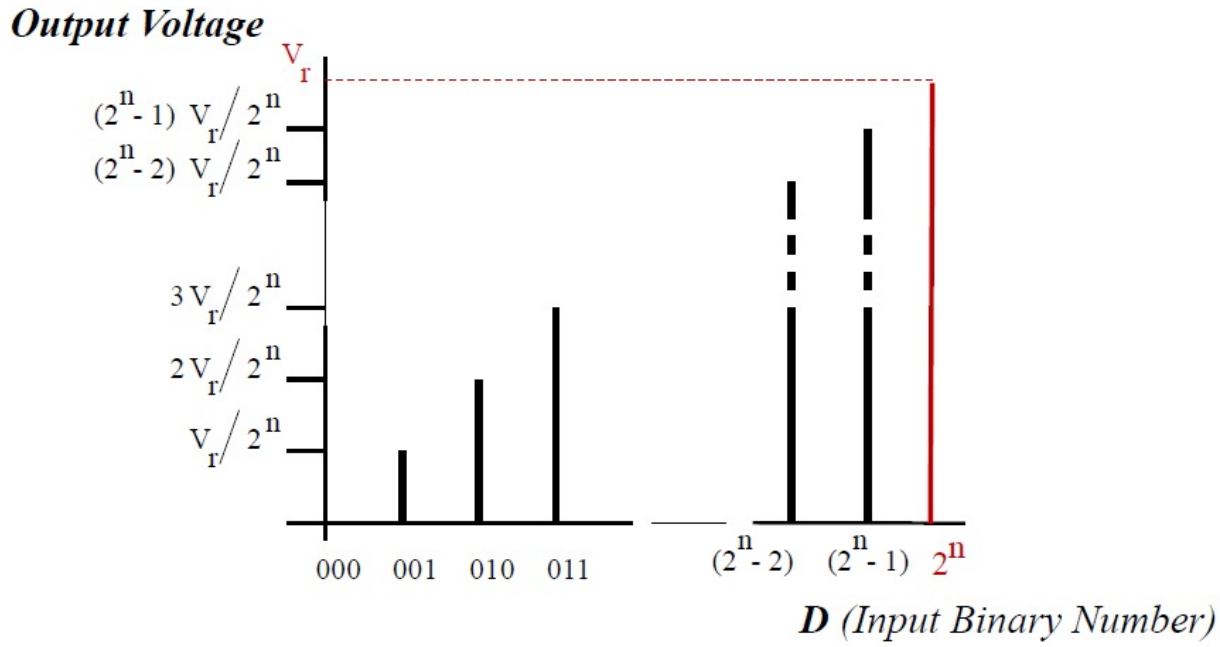
V_o = Analog Output

D = Digital input

n = n-bit of Digital Input (resolution)

V_r = Voltage Reference

คุณสมบัติของตัวแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอก



Thingcontrol board ใช้ ESP32-WROOM-32 เป็น MCU ที่มี DAC มาให้ 2 Channel คือ Ch.1 ที่ GPIO25 และ Ch.2 ที่ GPIO26 โดยทั้ง 2 ช่องนี้เป็น DAC ที่มีความละเอียด 8 bit (0 ถึง 255 = 256 ระดับ) และใช้ไฟเลี้ยงที่ 3.3 V ซึ่งเท่ากับ Voltage Reference ดังนั้นความละเอียดของแต่ขั้น 3.3/256 = 12.9 mV

ตัวอย่าง

Digital Input

0000 0000 = 0

0000 0001 = 1

0000 0010 = 2

……

1111 1110 = 254

1111 1111 = 255

Analog Output

V_o = 0 – 255/255 x 3.3

= 0/255 * 3.3 = 0 V

= 1/255 * 3.3 = 0.0129 V

= 2/255 * 3.3 = 0.0258 V

……..

= 254/255 * 3.3 = 3.2870 V

= 255/255 * 3.3 = 3.3 V

คำสั่งในเอาท์พุตแบบอนาลอก DAC

dacWrite(pin, value) เป็นการสั่งให้ขาพอร์ทที่ระบุไว้ มีค่าของ digital input ของ DAC

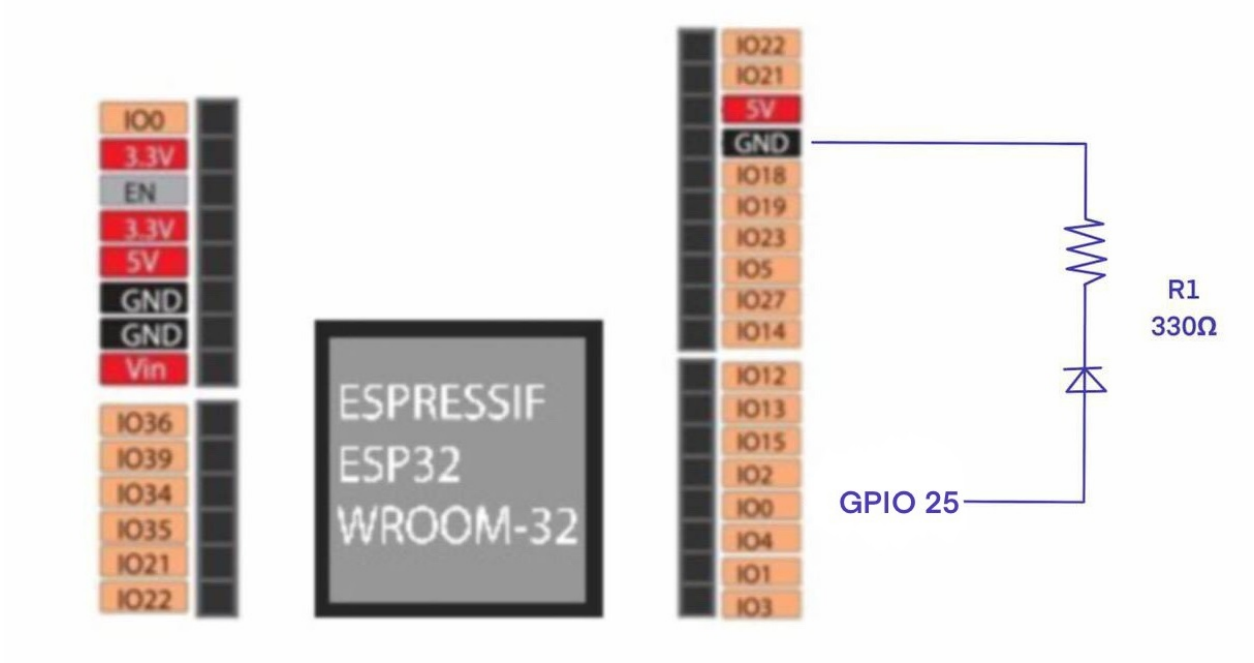
pin - หมายเลขของ DAC Channel 1 ที่ GPIO25 และ Ch2 ที่ GPIO26 ของ thingcontrol board

value – ค่าของ digital input ของ DAC ขึ้นอยู่กับความละเอียดของ DAC

delay(value) เป็นการหน่วงเวลาของโปรแกรมตามเวลาที่กำหนดมีหน่วยเป็นมิลลิวินาที

value – มีค่าเป็น, milliseconds (1 second = 1000 milliseconds)

รูปแบบการต่อใช้งานของ DAC

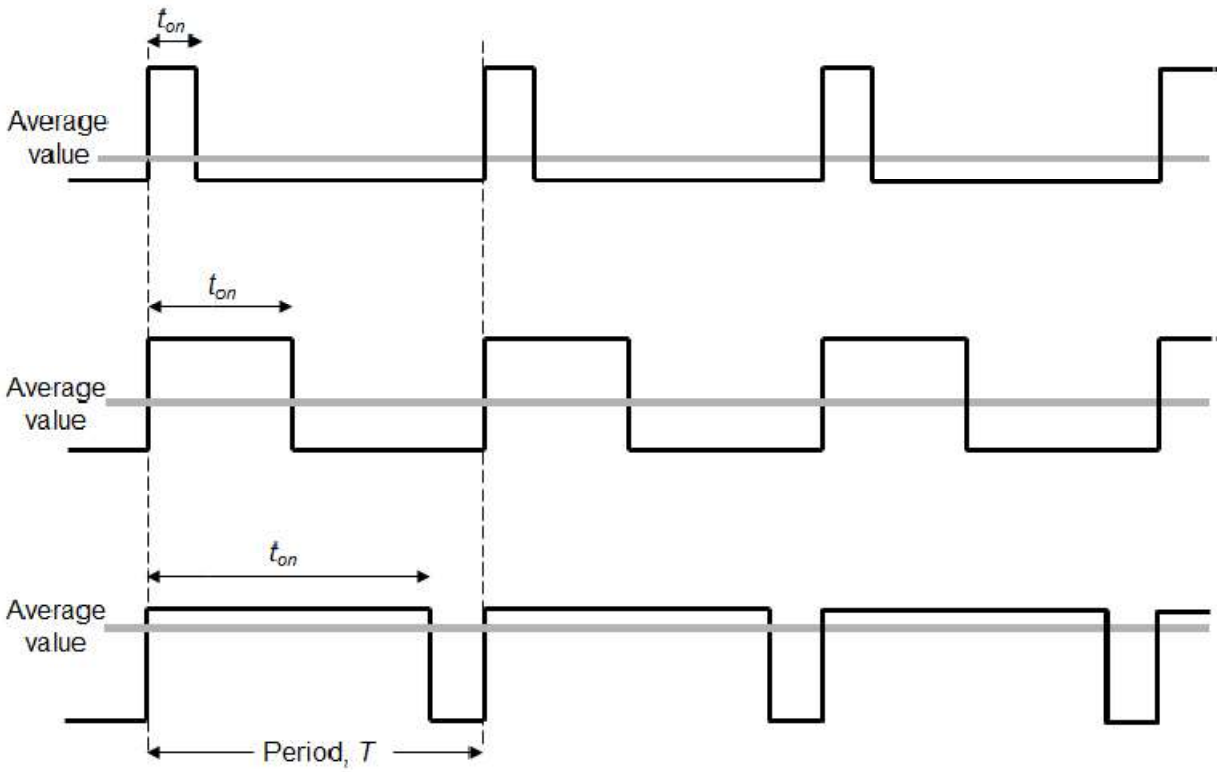


ตัวอย่างโปรแกรม

```
#define DAC1 25


int dacOut = 0;

int dacAmount = 5;


void setup() {

  Serial.begin(115200);

}


void loop() {
```

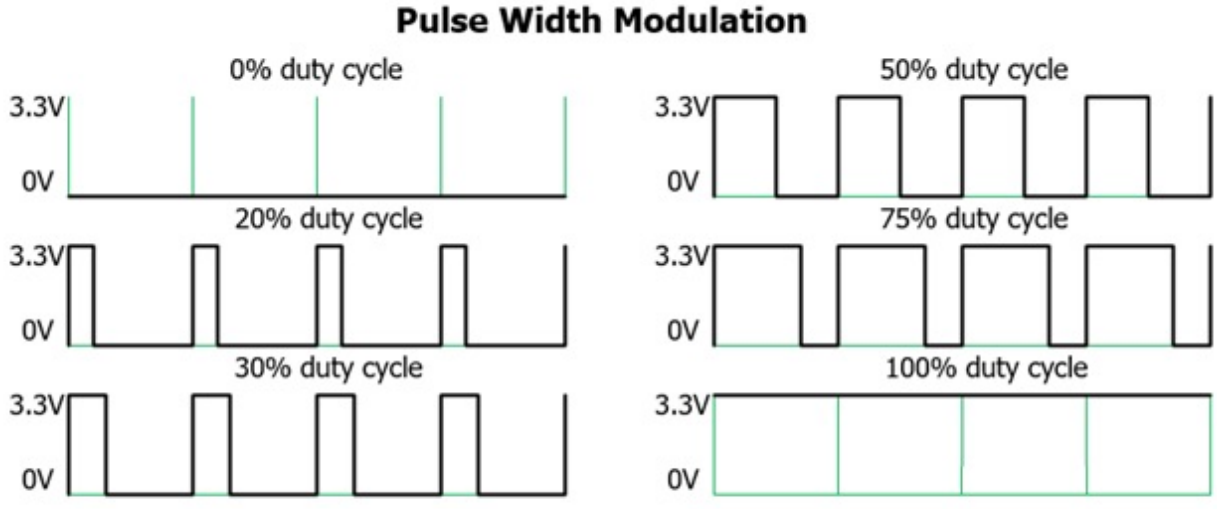
```
  Serial.print(dacOut);
  Serial.print("   ");
  Serial.println(dacOut * 0.0129);

  dacWrite(DAC1, dacOut);   //dacout 255= 3.3V,  128=1.65V

  dacOut = dacOut + dacAmount;
  if (dacOut <= 0 || dacOut >= 255) {
    dacAmount = -dacAmount;
  }
  delay(50);
}
```

2. การสร้างสัญญาณพัลส์วิดท์มอดูเลชัน (Pulse Width Modulation) หรือ PWM เป็นเทคนิคอีกแบบของการแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอก สัญญาณที่ถูกสร้างด้วยเทคนิคการสร้างระดับแรงดันอนาล็อกจากค่าเฉลี่ยแรงดันของสัญญาณดิจิตอล โดยการควบคุมทางความกว้างของพัลส์ (Pulse Width) โดยความกว้างของพัลส์จะเปลี่ยนแปลงให้สอดคล้องกับแรงดันอนาลอกที่ต้องการ แต่ความถี่ยังคงเดิม อัตราส่วนระหว่างช่วงคลื่นที่มีแรงดันต่อคาบเวลาคลื่นเรียกว่า Duty cycle ดังรูป



Duty Cycle = pulse on time/pluse period x 100%

V_o = Duty Cycle * Voltage



V_o = Duty Cycle * Voltage

= 0 * 3.3 = 0 V

= 25/100 * 3.3 = 0.825 V

= 50/100 * 3.3 = 1.65 V

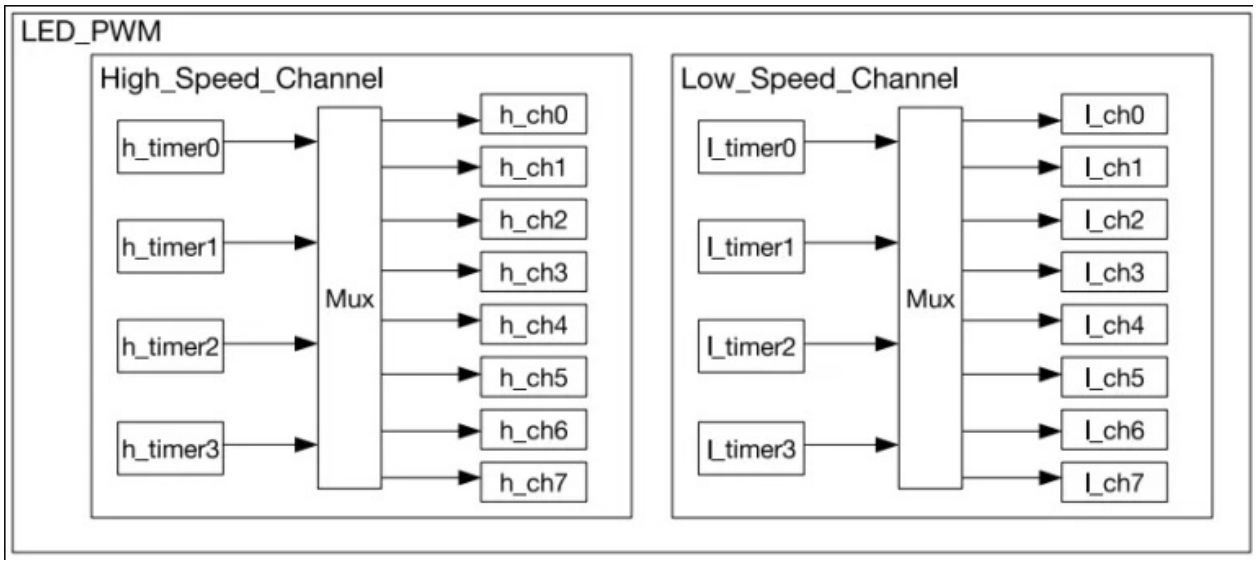
= 75/100 * 3.3 = 2.475 V

= 100/100 * 3.3 = 3.3 V

Thingcontrol board ใช้ ESP32-WROOM-32 เป็น MCU ที่มี PWM Hardware จำนวน 16 Channel (ไม่ใช่จำนวน Pin) ซึ่งเราสามารถโปรแกรมให้ GPIO Pin ไหนก็ได้ ให้เป็น PWM โดยจะแบ่งเป็น

1. 8 High-speed channel
2. 8 Low-speed channel

ESP32 PWM hardware diagram



คำสั่งในเอาท์พุตแบบอนาลอก PWM

ledcSetup(channel,freq,resolution) เป็นการกำหนด Channel ของ PWM

channel – หมายเลขของ channel สามารถมีค่าได้ 0 – 15 (16 Channel)

freq - ค่าความถี่ที่ใช้สร้างสัญญาณ PWM

resolution - resolution ค่าความละเอียดของ Duty cycle 1-16 bit เช่นถ้าใช้ 8 bit ค่า Duty cycle ที่กำหนดจะมีค่า 0-255 หมายถึง 0-100%

ledcAttachPin(GPIO, channel) เป็นการกำหนดหมายเลข GPIO pin ให้กับ Channel

GPIO - หมายเลข GPIO pin ที่ต้องการใช้

Channel – หมายเลข Channel ที่เลือกใช้งานกับ หมายเลข GPIO pin

ledcWrite(channel, dutycycle) เป็นการสั่งให้หมายเลข Channel ที่ระบุไว้ มีค่าของ duty cycle ของ PWM ที่ต้องการ

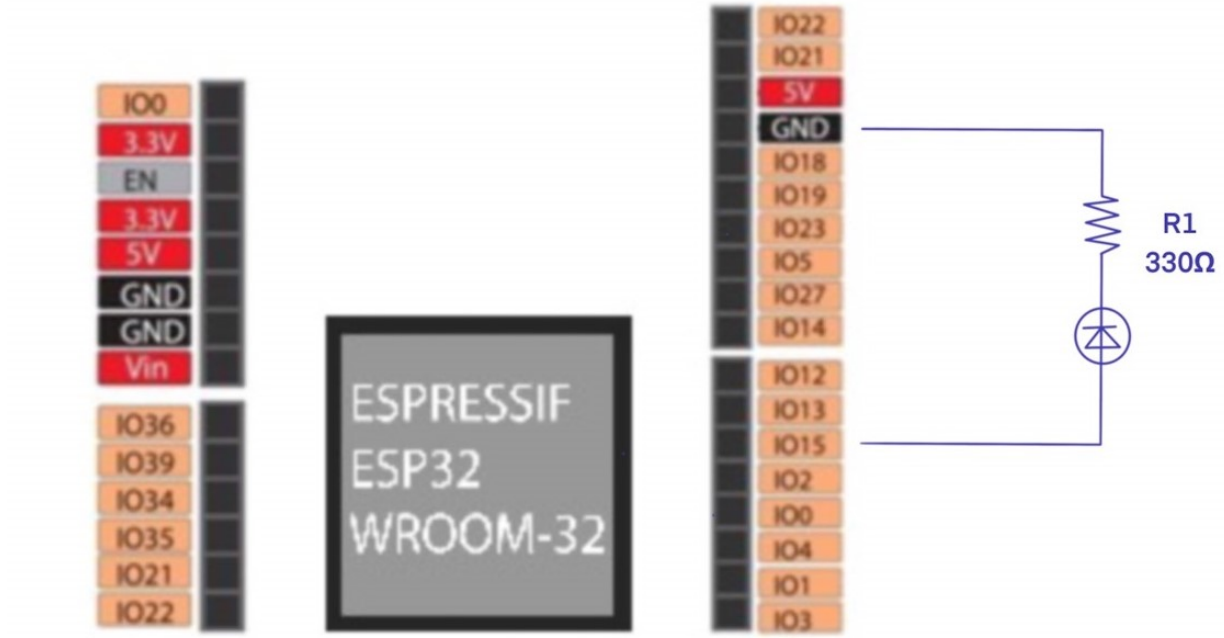
channel - หมายเลข channel ที่ต้องการ

dutycycle – ค่าของ Duty cycle ที่ต้องการสร้างขึ้น

delay(value) เป็นการหน่วงเวลาของโปรแกรมตามเวลาที่กำหนดมีหน่วยเป็นมิลลิวินาที

value – มีค่าเป็น, milliseconds (1 second = 1000 milliseconds)

รูปแบบการต่อใช้งานของ PWM



ตัวอย่างโปรแกรม

```
// the number of the LED pin

const int ledPin = 15;  // 16 corresponds to GPIO15



// setting PWM properties

const int freq = 5000;

const int ledChannel = 0;

const int resolution = 8;
```

```
void setup(){
  // configure LED PWM functionalitites
  ledcSetup(ledChannel, freq, resolution);

  // attach the channel to the GPIO to be controlled
  ledcAttachPin(ledPin, ledChannel);
}

void loop(){
  // increase the LED brightness
  for(int dutyCycle = 0; dutyCycle <= 255; dutyCycle++){
    // changing the LED brightness with PWM
    ledcWrite(ledChannel, dutyCycle);
    delay(15);
  }

  // decrease the LED brightness
  for(int dutyCycle = 255; dutyCycle >= 0; dutyCycle--){
    // changing the LED brightness with PWM
    ledcWrite(ledChannel, dutyCycle);
```

```
    delay(15);
  }
}
```

คำถามท้ายใบความรู้ที่ 7

1.จงอธิบายหลักการทำงานของตัวแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอก (Digital to Analog Converter)

2. จงอธิบายหลักการทำงานสัญญาณพัลส์วิดท์มอดูเลชัน (Pulse Width Modulation)

3. จงบอกคำสั่งที่ใช้การแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอก ด้วยวิธีการสร้างสัญญาณพัลส์วิดท์มอดูเลชัน (Pulse Width Modulation)

3.1 ledcSetup(channel,freq,resolution)

3.2 ledcAttachPin(GPIO, channel)

3.3 ledcWrite(channel, dutycycle)