

# Tipos de señales

Verónica E. Arriola-Rios

Robótica móvil

21 de agosto de 2025

# Temas

## 1 Modulación por ancho de pulso (PWM)

- Definiciones
- Aplicaciones
- Ejercicio

## 2 Señales analógicas

- Discretización y cuantización
- Ejercicio

# Temas

## 1 Modulación por ancho de pulso (PWM)

- Definiciones
- Aplicaciones
- Ejercicio

## 2 Señales analógicas

- Discretización y cuantización
- Ejercicio

# PWM (*Pulse Width Modulation*)

- Se puede traducir al español como *ancho de pulso modulado*.
- El pulso está formado por una señal de onda cuadrada periódica.
- A lo largo del tiempo la señal varía entre dos valores de tensión.

$T_{on}$  *Time on*, el tiempo que la señal se encuentra en el nivel alto.

$T_{off}$  *Time off*, el tiempo que está en el nivel bajo.

**Periodo (T)** la suma de  $T_{on}$  y  $T_{off}$ .

**Frecuencia** el inverso del periodo  $1/T$ .



**Figura:** Onda cuadrada periódica. Alterna entre un nivel alto de 5V y uno bajo GND.

# Ciclo de trabajo

## Definición (Ciclo de trabajo)

El *ciclo de trabajo* (*Duty Cycle*) es el tiempo que la señal está activa frente al tiempo que la señal está apagada. (Santos 2018)

$$DC = \frac{T_{on}}{T} \quad (1)$$

- Con un ciclo de trabajo del 50 % la señal estará activa la mitad del tiempo, mientras que la otra mitad del tiempo estará apagada.
- Si el ciclo de trabajo es del 100 %, la señal estará activa todo el tiempo.

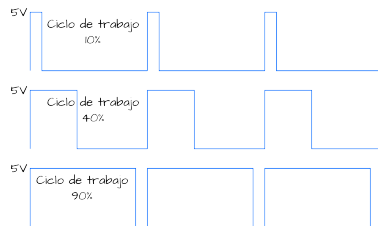


Figura: Señales PWM 10-40-90

# Variación de ancho de pulso

- La variación de ancho de pulso consiste en variar el ancho del pulso en el nivel alto  $T_{on}$  mientras  $T$  permanece fijo.

# Temas

## 1 Modulación por ancho de pulso (PWM)

- Definiciones
- Aplicaciones
- Ejercicio

## 2 Señales analógicas

- Discretización y cuantización
- Ejercicio

# Aplicaciones

- Variar la energía recibida por un dispositivo electrónico variando rápidamente la energía que éste recibe al cambiar entre apagado y encendido.
  - Controlar la velocidad de un motor de corriente continua, como el de los ventiladores de una PC.
  - Controlar la intensidad de la iluminación por leds.
- Comunicación entre redes.
- En amplificadores, teniendo una eficiencia más alta que otras alternativas. (Santos 2018)
- Enviar una señal PWM que varia su ciclo de trabajo a un altavoz o zumbador para generar sonidos y melodías de manera sencilla y rápida. (Gómez 2017)



# Temas

## 1 Modulación por ancho de pulso (PWM)

- Definiciones
- Aplicaciones
- Ejercicio

## 2 Señales analógicas

- Discretización y cuantización
- Ejercicio

# Circuito

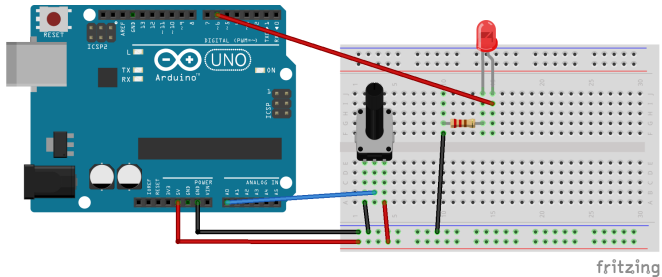


Figura: (Diagrama original)

Material:

- Potenciómetro de no más de 10K.
- Resistencia ( $\sim 330\Omega$ )

<https://moviltronics.com/controlar-el-brillo-de-un-led-con-un-potenciometro-arduino/>

# LEDs

Valores estándares de voltaje requerido por LEDs de diferentes colores:

Color	Tensión umbral
Rojo	1.9V
Amarillo	1.7V a 2V
Verde	2.4 V
Naranja	2.4 V
Blanco	3.4 V
Azul	2.4

Pueden soportar una corriente de 20mA aproximadamente.  
Resistencia necesaria para el circuito:

$$V - V_{LED} = RI$$

$$R = \frac{5V - 1.8V}{0.02A} = 160\Omega$$

# Cómo generar una señal PWM

```
1  const int LED_PWD = 6;
2  const int POT_PIN = A0;
3  int value = 0;
4
5  void setup() {
6      Serial.begin(9600);
7      pinMode(LED_PWD, OUTPUT);
8  }
9
10 void loop() {
11     value = analogRead(POT_PIN);
12     Serial.print("Pot:");
13     Serial.println(value);
14     value = map(value, 0, 1024, 0, 255);
15     analogWrite(LED_PWD, value);
16 }
```

# Temas

## 1 Modulación por ancho de pulso (PWM)

- Definiciones
- Aplicaciones
- Ejercicio

## 2 Señales analógicas

- Discretización y cuantización
- Ejercicio

# Temas

## 1 Modulación por ancho de pulso (PWM)

- Definiciones
- Aplicaciones
- Ejercicio

## 2 Señales analógicas

- Discretización y cuantización
- Ejercicio

# Conversión analógica digital

Stage 1

Stage 2

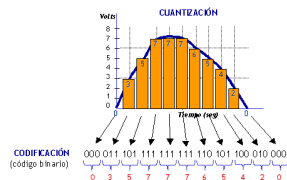
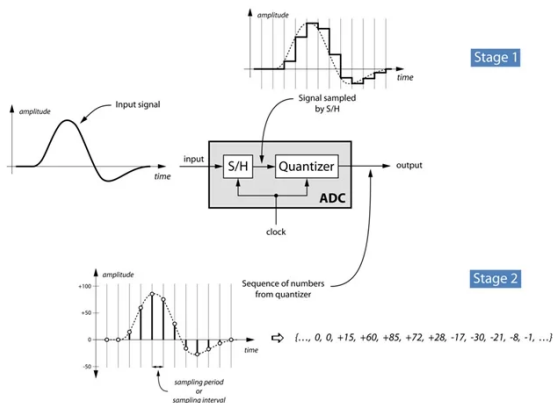


Figura: jecrespom 2017

# DAC en Arduino I

El microcontrolador de Arduino UNO contiene internamente un **único** conversor analógico a digital con:

- 6 canales
- resolución de 10 bits (enteros entre 0 y 1023)

Uso:

- Sólo se puede leer el valor de un puerto analógico a la vez, porque sólo hay un DAC.
- Se aconseja que después de manipular pines analógicos (en modo digital), añadir un pequeño retraso antes de usar `analogRead()` para leer otros pines analógicos, para evitar ruido eléctrico y la introducción de jitter en el sistema analógico.



# DAC en Arduino II

- En lugar de leer voltajes cuyo máximo es 5V, es posible utilizar el pin AREF para voltajes más pequeños (ej: 3.3V). Esto ofrece mayor resolución pues:

$$V = \frac{V_{\text{ref}}}{1024} * \text{lectura} \quad \text{lectura} \in [0, 1023]$$

Para ello usar `AnalogReference()`

# Convertidor Digital Analógico (DAC)

- Sólo lo tienen Arduino DUE, Zero o MKR1000.
- Se usan para los altavoces, amplificadores para producir sonido. Ejemplo de la transmisión de la voz por la líneas telefónicas.
- Como los pulsos PWM sólo mantienen el voltaje máximo durante un porcentaje del ciclo de trabajo, que ocurre a alta frecuencia, se utilizan para simular una señal analógica en la que se emite un voltaje igual al voltaje promedio.

# Temas

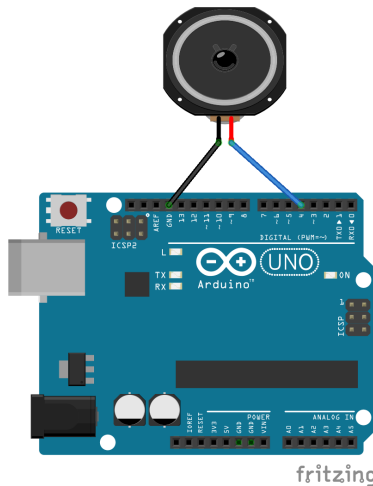
## 1 Modulación por ancho de pulso (PWM)

- Definiciones
- Aplicaciones
- Ejercicio

## 2 Señales analógicas

- Discretización y cuantización
- Ejercicio

# Circuito



# Arturito I

## Referencias:

- <https://www.instructables.com/R2D2-Sound-Generator/>
- [https://github.com/veroarriola/paquito-zero/tree/main/bosquejo\\_completo/Speak](https://github.com/veroarriola/paquito-zero/tree/main/bosquejo_completo/Speak)

```
1  const int PIN_LED = 13;
2  const int PIN_BUZZER = 4;
3
4  void setup() {
5      pinMode(PIN_BUZZER, OUTPUT);
6      pinMode(PIN_LED, OUTPUT);
7      randomSeed(analogRead(0));
8  }
9
10 void whistle(int baseFreq, bool up, int timesFirst, int timesSecond) {
11     int dir = (up) ? 1 : -1;
12     digitalWrite(PIN_LED, HIGH);
13     for (int i = 0; i <= timesFirst; i++){
14         tone(PIN_BUZZER, baseFreq + (-dir * i * 2));
15         delay(random(.9, 2));
16     }
17     digitalWrite(PIN_LED, LOW);
18     for (int i = 0; i <= timesSecond; i++){
19         tone(PIN_BUZZER, baseFreq + (dir * i * 10));
20         delay(random(.9, 2));
21     }
22 }
23
24 void phrase(bool up) {
```





# Arturito II

```
25     int baseFreq;
26     int timesFirst;
27     int timesSecond;
28
29     if (up) {
30         baseFreq = random(700,1000);
31         timesFirst = random(100,400);
32         timesSecond = random(100,1000);
33     } else {
34         baseFreq = random(2000,3000);
35         timesFirst = random(200,1000);
36         timesSecond = random(50,150);
37     }
38     whistle(baseFreq, up, timesFirst, timesSecond);
39 }
40
41 void babble() {
42     int K = 2000;
43     for (int i = 0; i <= random(3, 9); i++){
44         digitalWrite(PIN_LED, HIGH);
45         tone(PIN_BUZZER, K + random(-1700, 0));
46         delay(random(70, 170));
47         digitalWrite(PIN_LED, LOW);
48         noTone(PIN_BUZZER);
49         delay(random(0, 30));
50     }
51 }
52
```

# Arturito III

```
53 void loop() {
54     switch (random(1,7)) {
55         case 1: phrase(true); break;
56         case 2: phrase(false); break;
57         case 3: phrase(true); phrase(false); break;
58         case 4: phrase(true); phrase(false); phrase(true); break;
59         case 5: phrase(true); phrase(false); phrase(true); phrase(false); phrase(true)
        ->); break;
60         case 6: phrase(false); phrase(true); phrase(false); break;
61     }
62     babble();
63     noTone(PIN_BUZZER);
64     delay(random(2000, 4000));
65 }
```

# Referencias I

-  Gómez, Enrique (dic. de 2017). *Qué es PWM y para qué sirve*. Español. Rincón ingenieril. URL:  
<https://www.rinconingenieril.es/que-es-pwm-y-para-que-sirve/>.
-  jecrespom (sep. de 2017). *Puertos Analógicos Arduino Avanzado*. URL:  
<https://aprendiendoarduino.wordpress.com/2017/09/05/puertos-analogicos-arduino-avanzado/>.
-  Moviltronics (s.f.). *Controlar el brillo de un LED con un potenciómetro arduino*. URL: <https://hetpro-store.com/TUTORIALES/resistencia-de-led/>.
-  Santos, Manuel (mar. de 2018). *GeneralPWM: qué usos tiene en ventiladores y luces LED de nuestro PC*. Español. Hard Zone. URL:  
<https://hardzone.es/2018/03/11/uso-pwm-pc/>.