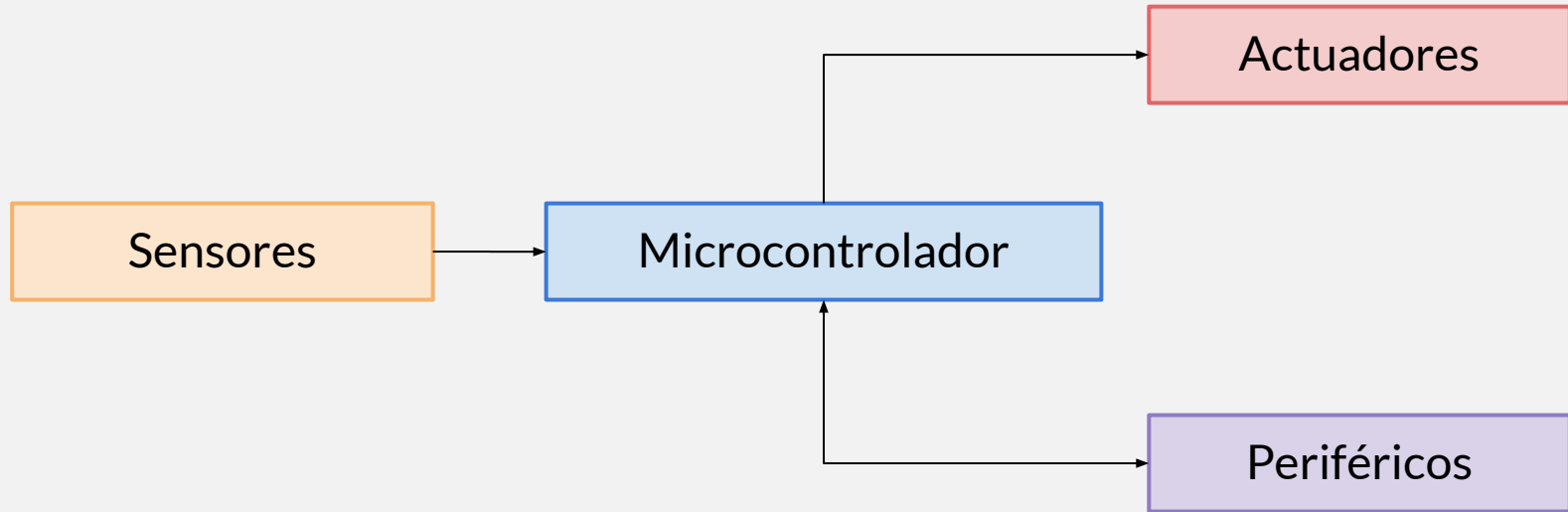




# Sensores, Actuadores y Periféricos



# Sensores, Actuadores y Periféricos





- **Conexiones externas**

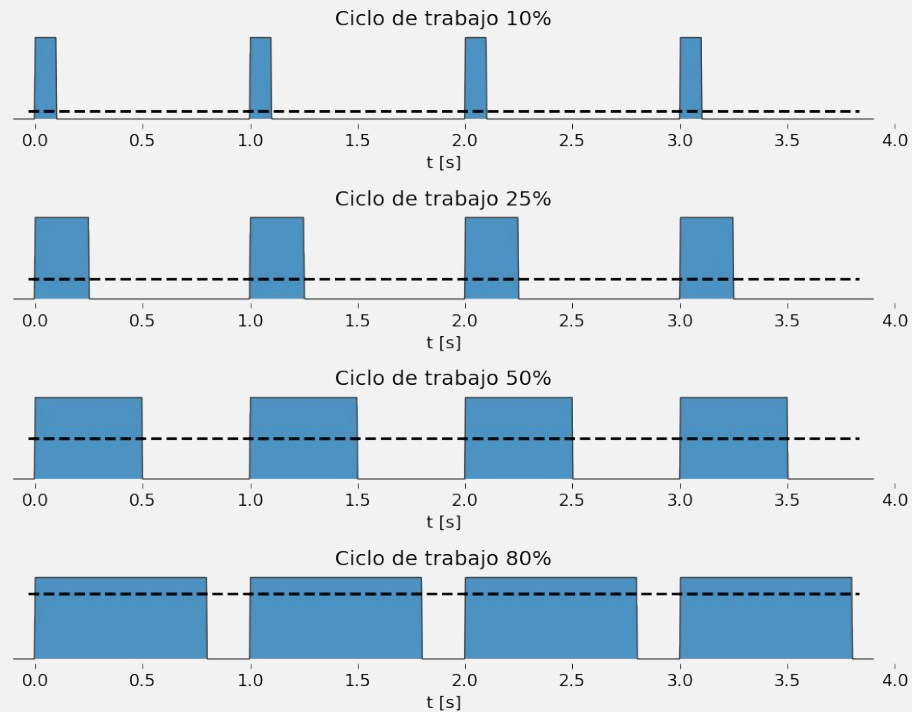
- Entrada/salida digital (1 o más bits)
- PWM
- Entrada/salida analógica
- Comunicación
  - UART
  - I2C
  - SPI



- Modulación de ancho de pulso
- Onda pulsada de frecuencia fija y ciclo de trabajo variable
- La información se transmite en la duración (o ancho) del pulso
- Aplicaciones:
  - Brillo regulable para LEDs
  - Control de motor de continua
  - Control de giro de servomotor



# PWM





- Pines disponibles en kit de desarrollo:
  - LEDs: GPIO0, GPIO2, GPIO15
  - Servomotor: GPIO14
- Configuración:

```
1 from machine import Pin, PWM
2
3 pwm = PWM(Pin(0), freq=10000)
4
5 pwm.duty(512)
```

—————→ Ciclo de trabajo: 0 – 1023



# PWM – Control de motor de continua

- Valor medio determina la velocidad de giro
- Para ciclo de trabajo 0: mínima velocidad (valor medio nulo)
- Para ciclo de trabajo 1023: máxima velocidad (valor medio máximo)
- Interfaz de potencia: puente H (circuitería adicional)



# PWM – Control de servomotor

- Se configura a 50Hz (pulsos separados 20 ms)
- Para la posición 0°: pulso de 1ms (ciclo de trabajo 51)
- Para la posición 90°: pulso de 1.5ms (ciclo de trabajo 76)
- Para la posición 180°: pulso de 2ms (ciclo de trabajo 102)





# Entrada/Salida analógica

- Conversor analógico-digital de 12 bits
- Conversor digital-analógico de 8 bits
- Aplicaciones:
  - Leer tensiones variables de sensores como temperaturas, luminosidad, etc.
  - Generar formas de onda arbitrarias o valores de tensión específicos



# Configuración ADC

```
1 from machine import Pin, ADC
2
3 adc = ADC(Pin(35), atten=ADC.ATTN_11DB)
4
5 lectura = adc.read_uv()
6 print(f"{lectura/1000} mV")
```

Factor de atenuación:

- ADC.ATTN\_0DB: 100mV ~ 950mV
- ADC.ATTN\_2\_5DB: 100mv ~ 1250mV
- ADC.ATTN\_6DB: 150mV ~ 1750mV
- ADC.ATTN\_11DB: 150mV ~ 2450mV



# Configuración DAC

```
1 from machine import Pin, DAC
2
3 dac = DAC(Pin(25))
4
5 dac.write(127)
```

→ Amplitud entre 0 - 255



# Protocolos

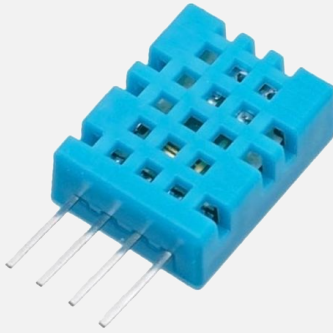
- UART: comunicación serie asincrónica
- I2C: comunicación serie sincrónica de baja velocidad
- SPI: comunicación serie sincrónica de alta velocidad

## Aplicaciones:

- UART: terminal en PC usando conversor UART a USB
- I2C: pantallas LCD, sensores varios
- SPI: tarjetas SD, pantallas LCD



# Otros sensores y actuadores



Sensor de temperatura y  
humedad DHT11/22



LED RGB NeoPixel (ws2812b)



# Configuración DHT11/22

```
1 from dht import DHT11
2 from machine import Pin
3
4 dht = DHT11(Pin(32))
5
6 dht.measure()
7 print(f"Temperatura: {dht.temperature()} °C")
8 print(f"Humedad: {dht.humidity()} %")
```



# Configuración NeoPixel

```
1 from neopixel import NeoPixel
2 from machine import Pin
3
4 np = NeoPixel(Pin(27), 3)
5
6 # Orden: R, G, B
7 np[0] = (255, 0, 0)
8 np[1] = (0, 255, 0)
9 np[2] = (0, 0, 255)
10 np.write()
```



- **Ejercicio 1**

- Hacer parpadear un LED de forma suave, variando su brillo

- **Ejercicio 2**

- Cambiar el brillo un LED según la posición del potenciómetro

- **Ejercicio 3**

- Cambiar el ángulo de giro del servomotor usando el potenciómetro





- **Ejercicio 4**

- Encender el LED verde, amarillo o rojo según la posición del potenciómetro. Entre 0 – 1V, verde. Entre 1 – 2V, amarillo. Entre 2 – 3V, rojo.

- **Ejercicio 5**

- Leer la temperatura del DHT11/22 y encender el LED verde si la temperatura está por debajo de 30°C. Si está por encima, encender el LED rojo