# Documentación de la Biblioteca 'machine' de MicroPython

La biblioteca 'machine' de MicroPython es una de las bibliotecas más importantes para interactuar con el hardware de la placa, como la ESP32 y otras placas compatibles. Esta biblioteca permite configurar y controlar pines GPIO, temporizadores, comunicación por bus (I2C, SPI, UART), y más.  
  
A continuación se presenta una lista completa de las clases y métodos de la biblioteca 'machine'.

## Clases de la Biblioteca 'machine':

1. Pin

2. Timer

3. ADC

4. DAC

5. PWM

6. RTC

7. UART

8. I2C

9. SPI

10. WDT

11. Freq

12. SDCard

13. USB

14. Network

15. Sleep

## Métodos de las Clases:

### Clase 'Pin':

\_\_init\_\_(pin, mode, pull)

value([val])

on()

off()

irq(trigger, handler)

name()

### Clase 'Timer':

init(period, mode, callback)

deinit()

callback()

### Clase 'ADC':

\_\_init\_\_(pin)

read()

read\_u16()

### Clase 'DAC':

write(value)

### Clase 'PWM':

\_\_init\_\_(pin, freq, duty)

duty([duty])

freq([freq])

### Clase 'RTC':

datetime([datetime])

memory()

### Clase 'UART':

\_\_init\_\_(id, baudrate, bits, parity, stop)

read([nbytes])

write(data)

any()

### Clase 'I2C':

\_\_init\_\_(id, scl, sda)

scan()

readfrom(address, nbytes)

writeto(address, buffer)

readfrom\_into(address, buffer)

writeto\_then\_readfrom(address, buffer\_out, buffer\_in)

### Clase 'SPI':

\_\_init\_\_(id, baudrate, polarity, phase, bits, firstbit)

write(buf)

read(nbytes)

write\_readinto(buf\_out, buf\_in)

### Clase 'WDT':

\_\_init\_\_(timeout)

feed()

deinit()

### Clase 'Freq':

set\_frequency(freq)

### Clase 'SDCard':

\_\_init\_\_(slot)

read\_blocks(block, count)

write\_blocks(block, count, buf)

### Clase 'USB':

Métodos específicos para manejar conexiones USB, dependiendo del modelo de la placa.

### Clase 'Network':

WLAN()

Ethernet()

Métodos para conectar, desconectar y gestionar la red.

### Clase 'Sleep':

sleep()

deepsleep()