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()