UART

> Página extraída de Federico Coca Guia de Trabajo de Microbit https://fgcoca.github.io/Guia-de-trabajo-para-microbit/conceptos/serie/ Licencia CC-BY-SA

En MicroPython es el módulo *uart* el que permite comunicarse a través de la interfaz serie entre la micro:bit y el ordenador. El módulo tiene diferentes funciones encomendadas a realizar diferentes tareas de comunicación serie, pero sobre todo hay una fundamental que es la de inicializar la micro:bit para trabajar.

```
microbit.uart.init(baudrate=9600, bits=8, parity=None, stop=1, *, tx=None, rx=None)
```

Inicializa la comunicación serie con los parámetros especificados en los pines *Tx* y *Rx* especificados. Hay que tener en cuenta que, para que la comunicación sea correcta, los parámetros deben ser los mismos en los dispositovos que se comunican. En la función tenemos:

- **baudrate**. Define la velocidad de la comunicación en baudios y puede ser: 9600, 14400, 19200 (2x9600), 28800 (14400x2), 38400 (2x19200), 57600 (28800x2) o 115200 (57600x2).
- **bits**. El parámetro *bits* define el tamaño de los bytes que se transmiten. Micro:bit solamente soporta 8 bits.
- **parity**. El parámetro *paridad* define la forma de comprobación de la paridad pudiendo valer: `none`, `microbit.uart.ODD` o `microbit.uart.EVEN`, lo que indica: ninguna, impar o par respectivamente. La paridad es una forma de comprobar que el dato transmisitido y el recibido coinciden.
- **stop**. Este parámetro indica el número de bits de parada, que en el caso de la micro:bit es uno.
- **Tx y Rx**. Son los pines de transmisión (Tx) y recepción (Rx) de la placa. Si no se especifican se utilizan los internos de USB/UART. Se puede especificar cualquier otro pin.

Notas:
- Inicializar la UART en pines diferentes a los establecidos por defecto para USB puede originar que la consola de Python deje de ser accesible, ya que utiliza el mismo hardware.

Copyright 2026

> Para recuperar la consola hay que reinicializar la UART sin pasar nada por Tx o Rx (o pasando None a estos argumentos). Es decir, llamar a `uart.init(115200)` es suficiente para restaurar la consola Python.
> - Las conexiones de dispositivos mediante Tx y Rx requieren "cruzar" los cables, de forma que el pin TX de la placa esté conectado al Rx del otro dispositivo y el pin Rx, con el pin Tx. De esta forma lo que uno transmite el otro lo escucha. También es imprescindible que los dispositivos tengan un GND común.

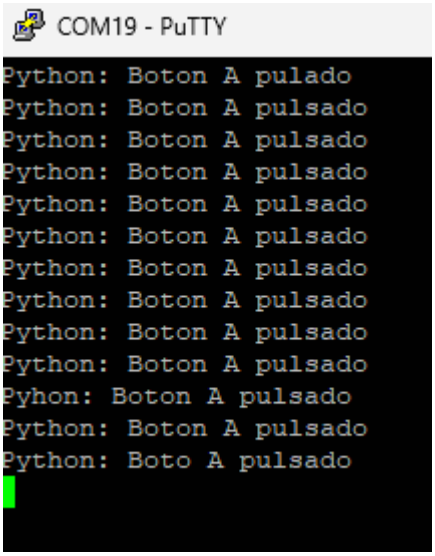
El ejemplo siguiente nos va a permitir comprobar la comunicación serie desde MicroPython. Grabamos el programa en la micro:bit.

```
from microbit import *


uart.init(115200,8,None,1)
while True:
    if button_a.was_pressed():
        #"\n" indica un salto de linea
        #\r" indica un retorno de carro
        #Con los dos se salta una linea y se lleva
        #el curso al principio
        uart.write("Python: Boton A pulsado\n\r")
```

Abrimos y configuramos PuTTY y la consola nos muestra:

> Página extraída de Federico Coca Guia de Trabajo de Microbit https://fgcoca.github.io/Guia-de-trabajo-para-microbit/conceptos/serie/ Licencia CC-BY-SA

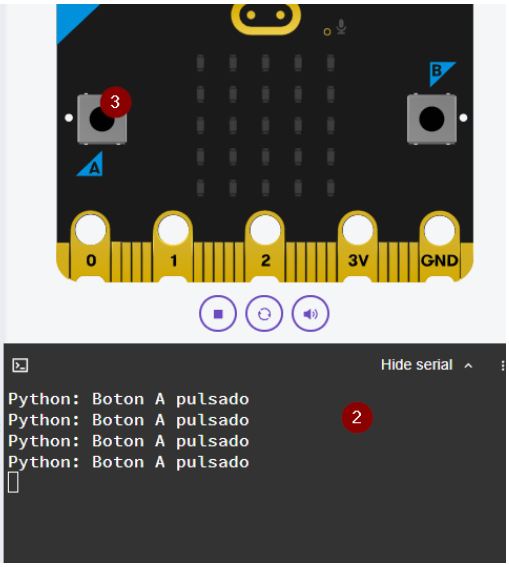


https://www.youtube.com/embed/44kSyEAdF6Q

En el simulador de Micropython también aparece

```
from microbit import *

uart.init(115200,8,None,1)
while True:
    if button_a.was_pressed():
        #"\n" indica un salto de linea
        #\r" indica un retorno de carro
        #Con los dos se salta una linea y se lleva
        #el curso al principio
        uart.write("Python: Boton A pulsado\n\r")
```



Revision #3
Created 2024-09-26 13:55:52 CEST by Juanjo Mejías

Copyright 2026

Updated 2024-09-26 14:37:48 CEST by Juanjo Mejías

Copyright 2026