

RASPBERRY ENVIA I PICO REP

CODI RASPBERRY ENVIANT 123\n

```
import serial
import time

# Configura el port UART
ser = serial.Serial('/dev/serial0', baudrate=4800, timeout=1) # Redueix el baudrate a 4800

# Dona temps al Pico per estar llest
time.sleep(2)

try:
    while True:
        # Esborra els buffers abans d'enviar
        ser.reset_input_buffer()
        ser.reset_output_buffer()

        # Envia un missatge al Pico
        message = "123\n" # Afegim un salt de línia per marcar el final del missatge
        try:
            ser.write(message.encode('utf-8'))
            print(f"Enviat: {message.strip()}")
        except serial.SerialException as e:
            print(f"Error d'escriptura: {e}")
            ser.close()
            time.sleep(1)
            ser.open()

        # Dorm una mica per evitar saturació al buffer del Pico
        time.sleep(2)
except KeyboardInterrupt:
    print("Interromput per l'usuari.")
    ser.close()
```

CODI PICO REBENT 123\n

```
from machine import UART, Pin

import utime

# Configura UART al Pico

uart = UART(0, baudrate=4800, tx=Pin(0), rx=Pin(1)) # Baudrate ajustat a 4800

# Funció per netejar el buffer inicial

def clear_uart_buffer():

    while uart.any():

        uart.read() # Esborra qualsevol dada restant

# Esborrem el buffer inicial abans de començar

clear_uart_buffer()

while True:

    if uart.any(): # Comprova si hi ha dades entrants

        try:

            # Llegeix totes les dades disponibles

            data = uart.read()

            print(f"Bytes rebuts: {data}") # Mostra els bytes crus

            # Filtra bytes vàlids (ASCII entre 32 i 127)

            filtered_data = bytes([b for b in data if 32 <= b <= 127])

            if filtered_data:

                try:

                    decoded_data = filtered_data.decode('utf-8').strip()

                    print(f"Dades decodificades: {decoded_data}")

                except UnicodeDecodeError as e:

                    print(f"Error de decodificació: {e}")

            except Exception as e:

                print(f"Error en manejar les dades: {e}")

        else:

            # Mostra un missatge si no hi ha dades

            print("Esperant dades...")

    utime.sleep(0.5) # Temps d'espera ajustat per evitar saturacions
```