

Hey ChatGPT, I am a fab academy student. Im using a Xiao RP2040 with a DC motor connected GPIO28. I want you to make an interface with Tkinter that has an on and an off button. When the on button is pressed, the motor should turn on. When the off button is pressed, the motor should turn off. The XIAO RP2040 is connected to my computer via a USB-C cable with data transmitting capabilities. I want the interface to communicate with the Xiao RP2040 via the usb-c cable.

You can definitely do this. Since the XIAO RP2040 appears as a serial device over USB, the easiest approach is:

1. Run a MicroPython program on the XIAO RP2040 that listens for serial commands.
2. Create a Tkinter GUI on your computer with ON and OFF buttons.
3. When the ON button is pressed, send "ON" over serial.
4. When the OFF button is pressed, send "OFF" over serial.
5. The RP2040 receives those commands and turns the motor on or off using GPIO28.

On the XIAO RP2040, you can use:

```
from machine import Pin, PWM
import sys
import select
```

```
motor = PWM(Pin(28))
motor.freq(2000)
```

```
while True:
```

```
    if sys.stdin in select.select([sys.stdin], [], [], 0)[0]:
        cmd = sys.stdin.readline().strip()
```

```
        if cmd == "ON":
            motor.duty_u16(65535)
        elif cmd == "OFF":
            motor.duty_u16(0)
```

Then, on your computer, you can create a Tkinter interface that uses the `pyserial` library to send "ON" and "OFF" commands over the USB-C serial connection.

Ok, I put that code onto the Xiao RP2040. It is connected to my computer via COM8. can you make the Tkinter interface now?

Sure! Since your XIAO RP2040 is connected on **COM8**, you can use `pyserial` together with Tkinter. The GUI will have two buttons, **ON** and **OFF**, which send serial commands to the RP2040 over the USB-C cable.

```
import serial
import tkinter as tk

# CHANGE THIS to your port!
ser = serial.Serial('COM8', 115200)

def motor_on():
    ser.write(b'ON\n')

def motor_off():
    ser.write(b'OFF\n')

root = tk.Tk()
root.title("Motor Control")

btn_on = tk.Button(root, text="ON", command=motor_on, width=15, height=3)
btn_on.pack(pady=10)

btn_off = tk.Button(root, text="OFF", command=motor_off, width=15, height=3)
btn_off.pack(pady=10)

root.mainloop()
```

This program opens COM8 at 115200 baud and sends "ON" or "OFF" whenever the corresponding button is pressed. The RP2040 code you already loaded will receive those commands and turn the motor on or off.

Ok, How do i run this code and what should I do with this?

Once you have the code saved, you can run it directly on your computer. Here's how:

1. **Make sure Python is installed** on your computer.
2. **Install the pyserial library** if you haven't already. Open Command Prompt and type:

```
pip install pyserial
```

3. **Save the code** in a file, for example:

```
motor_control.py
```

4. **Make sure your XIAO RP2040 is connected** and that it appears as **COM8**.
5. **Run the program** from Command Prompt:

```
python motor_control.py
```

6. A window titled "**Motor Control**" should appear with two buttons:
 - **ON** → sends "**ON**" to the XIAO RP2040, turning the motor on.
 - **OFF** → sends "**OFF**" to the XIAO RP2040, turning the motor off.

Folder structure

```
motor_control.py
```

That's all you need on the computer side.

Summary

XIAO RP2040

- Running the MicroPython code you already uploaded.
- Connected via USB-C.
- Listening for "**ON**" and "**OFF**" commands.

Computer

- Running `motor_control.py`.
- Sending commands over COM8 when the buttons are pressed.

When everything is running, pressing the buttons in the Tkinter window should control the motor attached to GPIO28.