```python
import tkinter as tk
from tkinter import messagebox
import serial
import threading
import queue
import time

import matplotlib.pyplot as plt
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg

# ================= SERIAL CONFIG =================
SERIAL_PORT = "COM3"
BAUD_RATE = 9600

try:
    ser = serial.Serial(SERIAL_PORT, BAUD_RATE, timeout=1)
except Exception as e:
    messagebox.showerror("Serial Error", str(e))
    raise SystemExit

# ================= GLOBALS =================
data_queue = queue.Queue()
running = True

# ================= SEND DATA =================
def send_data():
    data = entry.get().strip()
    if data:
        ser.write((data + "\n").encode())
        text_box.insert(tk.END, f"Sent: {data}\n")
        entry.delete(0, tk.END)

# ================= SERIAL THREAD =================
def read_serial():
    while running:
        try:
            line = ser.readline().decode(errors="ignore").strip()
            if line:
                data_queue.put(line)
        except:
            pass

# ================= GUI UPDATE =================
x_data, y_data = [], []
start_time = time.time()

def update_gui():
    while not data_queue.empty():
        line = data_queue.get()

        text_box.insert(tk.END, f"Received: {line}\n")
        text_box.see(tk.END)

        # Limit text lines
        if int(text_box.index('end-1c').split('.')[0]) > 200:
            text_box.delete("1.0", "2.0")

        try:
            value = float(line)
            x_data.append(time.time() - start_time)
            y_data.append(value)

            if len(x_data) > 100:
                x_data.pop(0)
                y_data.pop(0)

            line_plot.set_data(x_data, y_data)
            ax.relim()
            ax.autoscale_view()
            canvas.draw_idle()

        except ValueError:
            pass

    root.after(100, update_gui)

# ================= CLOSE HANDLER =================
def on_close():
    global running
    running = False
    time.sleep(0.2)
    ser.close()
    root.destroy()

# ================= TKINTER UI =================
root = tk.Tk()
root.title("Serial Communication with Live Plot")
root.geometry("900x600")
root.protocol("WM_DELETE_WINDOW", on_close)

top_frame = tk.Frame(root)
top_frame.pack(pady=10)

entry = tk.Entry(top_frame, width=30)
entry.pack(side=tk.LEFT)

send_button = tk.Button(top_frame, text="Send", command=send_data)
send_button.pack(side=tk.LEFT, padx=5)

text_box = tk.Text(root, height=10)
text_box.pack(padx=10, pady=10, fill=tk.X)

# ================= MATPLOTLIB =================
fig, ax = plt.subplots(figsize=(7, 4))
line_plot, = ax.plot([], [], lw=2)
ax.set_title("Live Serial Data")
ax.set_xlabel("Time (s)")
ax.set_ylabel("Value")

canvas = FigureCanvasTkAgg(fig, master=root)
canvas.get_tk_widget().pack(fill=tk.BOTH, expand=True)

# ================= START =================
threading.Thread(target=read_serial, daemon=True).start()
root.after(100, update_gui)
root.mainloop()
```