



04_Python_4_GUI_GiorgioCitterio_UmbertoColom

Lo scopo di questa attività è gestire tramite interfaccia grafica GUI di python il sensore e l'attuatore.

parte 1

Nella prima parte abbiamo studiato il funzionamento della libreria tkinter tramite un video.

parte 2

In questa parte dell'attività la GUI dovrà gestire l'attuatore, con 3 soluzioni diverse.

Soluzione 1: In questa prima soluzione tramite la GUI si potrà aumentare o diminuire la velocità del motore di 10 e si potrà scegliere la direzione, avanti o indietro.

codice python guiAttuatore1:

```
import tkinter as tk
import serial
import struct
ID=b"BE"
MITTENTE=b"M001"
DESTINATARIO=b"D031"
TIPO=b"A1"
VUOT0=b"-----"
direzione = b"A"
velocita = 0

arduino = serial.Serial('COM3', 9600)
window = tk.Tk()
window.geometry("265x100")
window.title("Controllo motore")
window.resizable(False, False)

def StampaVelocità(vel):
    velocitaText_Label = tk.Label(window, text = "Velocità = " + str(vel)).grid(row=2, column=0, pady=3)

def AumentaVelocita10():
    global velocita
    if velocita >= 250:
        velocita = 250
    else:
        velocita += 10
    StampaVelocità(velocita)
    v=str(velocita).zfill(3).encode()
    pack=struct.pack("2s 4s 4s 2s 1s 3s 16s",ID,MITTENTE,DESTINATARIO, TIPO, direzione, v, VUOT0)
    arduino.write(pack)
    print(pack)

def DiminuisciVelocita10():
    global velocita
    if velocita <= 0:
        velocita = 0
    else:
        velocita -= 10
    StampaVelocità(velocita)
    v=str(velocita).zfill(3).encode()
    pack=struct.pack("2s 4s 4s 2s 1s 3s 16s",ID,MITTENTE,DESTINATARIO, TIPO, direzione, v, VUOT0)
    arduino.write(pack)
    print(pack)

def DirezioneAvanti():
    global direzione
    direzione = b"A"
    StampaVelocità(velocita)
    v=str(velocita).zfill(3).encode()
    pack=struct.pack("2s 4s 4s 2s 1s 3s 16s",ID,MITTENTE,DESTINATARIO, TIPO, direzione, v, VUOT0)
    arduino.write(pack)
```

```

def DirezioneIndietro():
    global direzione
    direzione = b"I"
    StampaVelocità(velocita)
    v=str(velocita).zfill(3).encode()
    pack=struct.pack("2s 4s 4s 2s 1s 3s 16s",ID,MITTENTE,DESTINATARIO, TIPO, direzione, v, VUOTO)
    arduino.write(pack)

aumentaVel_button = tk.Button(text="Aumenta velocità di 10", command=AumentaVelocita10).grid(row=0, column=0, sticky="W")

dimiuisceVel_button = tk.Button(text="Dimiuisce velocità di 10", command=DiminuisciVelocita10).grid(row=0, column=1, sticky="E")

avantiDir_button = tk.Button(text="Avanti", command=DirezioneAvanti).grid(row=1, column=0, sticky="W")

indietroDir_button = tk.Button(text="Indietro", command=DirezioneIndietro).grid(row=1, column=1, sticky="E")

if __name__ == "__main__":
    window.mainloop()

```

Soluzione 2: In questa seconda soluzione tramite la GUI si potrà inserire la velocità del motore tra 0 e 255 e la si potrà inviare al motore premendo invio, inoltre si potrà decidere la direzione, avanti o indietro.

codice python guiAttuatore2:

```

import tkinter as tk
import serial
import struct
ID=b"BE"
MITTENTE=b"M001"
DESTINATARIO=b"D031"
TIPO=b"A1"
VUOTO=b"-----"
direzione = b"A"
velocita = 0

arduino = serial.Serial('COM3', 9600)
window = tk.Tk()
window.geometry("225x60")
window.title("Controllo motore")
window.resizable(False, False)

def DirezioneAvanti():
    global direzione
    global velocita
    direzione = b"A"
    velocita = entry1.get()
    if int(velocita) > 255:
        velocita = 255
        entry1.delete(0, tk.END)
        entry1.insert(0, "255")
    elif int(velocita) < 0:
        velocita = 0
        entry1.delete(0, tk.END)
        entry1.insert(0, "0")
    v = str(velocita).zfill(3).encode()
    pack=struct.pack("2s 4s 4s 2s 1s 3s 16s",ID,MITTENTE,DESTINATARIO, TIPO, direzione, v, VUOTO)
    print(pack)
    arduino.write(pack)

def DirezioneIndietro():
    global direzione
    global velocita
    direzione = b"I"
    velocita = entry1.get()
    if int(velocita)>255:
        velocita = 255
        entry1.delete(0, tk.END)
        entry1.insert(0, "255")
    elif int(velocita) < 0:
        velocita = 0
        entry1.delete(0, tk.END)
        entry1.insert(0, "0")
    v = str(velocita).zfill(3).encode()
    pack=struct.pack("2s 4s 4s 2s 1s 3s 16s",ID,MITTENTE,DESTINATARIO, TIPO, direzione, v, VUOTO)
    print(pack)
    arduino.write(pack)

```

```

def InviaVelocita(event):
    global direzione
    global velocita
    velocita = entry1.get()
    if int(velocita)>255:
        velocita = 255
        entry1.delete(0, tk.END)
        entry1.insert(0, "255")
    elif int(velocita) < 0:
        velocita = 0
        entry1.delete(0, tk.END)
        entry1.insert(0, "0")
    v = str(velocita).zfill(3).encode()
    pack=struct.pack("2s 4s 4s 2s 1s 3s 16s",ID,MITTENTE,DESTINATARIO, TIPO, direzione, v, VUOTO)
    print(pack)
    arduino.write(pack)

avantiDir_button = tk.Button(text="Avanti", command=DirezioneAvanti).grid(row=0, column=0)

indietroDir_button = tk.Button(text="Indietro", command=DirezioneIndietro).grid(row=0, column=2)

velocitaText_Label = tk.Label(window, text = "Velocità:").grid(row=1, column=0, pady=5)

entry1 = tk.Entry(window, width=20)
entry1.grid(row=1, column=1)

enterEvent = window.bind('<Return>', InviaVelocita)

if __name__ == "__main__":
    window.mainloop()

```

Soluzione 3: In questa terza soluzione tramite la GUI con un campo di tipo Scale si specificherà la velocità che se sarà negativa la direzione sarà indietro, se sarà positiva la direzione sarà avanti. Inviando la velocità relativa.

codice python guiAttuatore3:

```

import tkinter as tk
import serial
import time
import struct
ID=b"BE"
MITTENTE=b"M001"
DESTINATARIO=b"D031"
TIPO=b"A1"
VUOTO=b"-----"
direzione = b"A"

arduino = serial.Serial('COM3', 9600)
window = tk.Tk()
window.geometry("570x100")
window.title("Controllo motore")
window.resizable(False, False)

def update_lbl(val):
    if int(val)<0:
        direzione = b"I"
        vel = -int(val)+1
        v = str(vel).zfill(3).encode()
        pack=struct.pack("2s 4s 4s 2s 1s 3s 16s",ID,MITTENTE,DESTINATARIO, TIPO, direzione, v, VUOTO)
        print(pack)
        arduino.write(pack)
    else:
        direzione = b"A"
        v = str(val).zfill(3).encode()
        pack=struct.pack("2s 4s 4s 2s 1s 3s 16s",ID,MITTENTE,DESTINATARIO, TIPO, direzione, v, VUOTO)
        print(pack)
        arduino.write(pack)

velocitaText_Label = tk.Label(window, text = "Velocità:").grid(row=0, column=0, pady=4)

scale = tk.Scale(window, orient="horizontal", length=510, from_=-255.0, to=255.0, command=update_lbl, cursor="target")
scale.grid(column=1, row=0)
scale.set(0)

if __name__ == "__main__":
    window.mainloop()

```

parte 3

In questa terza parte la GUI dovrà controllare la presenza di dati provenienti dal sensore con una frequenza superiore a quella dell'invio da parte del sensore.

Soluzione 1: In questa prima soluzione nella GUI si dovrà visualizzare il dato proveniente dal sensore.

codice python guiSensore1:

```
import tkinter as tk
import serial
import struct
IDCORRETTO = "BE"
DESTINATARIOCORRETTO = "D031"

arduino = serial.Serial('COM3', 9600)
window = tk.Tk()
window.geometry("225x60")
window.title("Controllo sensore")
window.resizable(False, False)

def StampaValori(val):
    valoreText_Label = tk.Label(window, text = "Valore sensore = " + str(val), font=('Courier New', 10)).grid(row=1, column=0, pady=13)

def funz(window):
    val = arduino.read(32)
    pack = struct.unpack("2s 4s 4s 2s 4s 16s", val)
    print(pack)
    id=pack[0].decode()
    mittente=pack[1].decode()
    destinatario=pack[2].decode()
    tipo=pack[3].decode()
    valoreSensore=pack[4].decode()
    vuoto=pack[5].decode()
    if (id==IDCORRETTO)and(destinatario==DESTINATARIOCORRETTO):
        print("id e destinatario corretti")
        print(valoreSensore)
        s=int(valoreSensore)
        StampaValori(valoreSensore)
    else:
        print("pacchetto scartato")
    window.after(1000, funz, window)

window.after(1000, funz, window)

if __name__ == "__main__":
    window.mainloop()
```

Soluzione 2: In questa seconda soluzione nella GUI bisognerà visualizzare un rettangolo che varierà la sua dimensione a seconda del valore del sensore, la dimensione sarà direttamente proporzionale al valore del sensore.

codice python guiSensore2:

```
import tkinter as tk
import serial
import struct
IDCORRETTO = "BE"
DESTINATARIOCORRETTO = "D031"

arduino = serial.Serial('COM3', 9600)
window = tk.Tk()
C = tk.Canvas(window, bg="purple", height=250, width=255)
C.pack()

rettangolo = C.create_rectangle(0, 0, 0, 0)

def funz(window):
    global rettangolo
    C.delete(rettangolo)
    val = arduino.read(32)
    pack = struct.unpack("2s 4s 4s 2s 4s 16s", val)
    print(pack)
```

```

id=pack[0].decode()
mittente=pack[1].decode()
destinatario=pack[2].decode()
tipo=pack[3].decode()
valoreSensore=pack[4].decode()
vuoto=pack[5].decode()
if (id==IDCORRETTO)and(destinatario==DESTINATARIOCORRETTO):
    print("id e destinatario corretti, valore del sensore = "+valoreSensore)
    v = int(valoreSensore)
    coordinate = 0, 70, v/4, 150
    rettangolo = C.create_rectangle(coordinate, fill="yellow")
else:
    print("pacchetto scartato")

    window.after(1000, funz, window)

window.after(1000, funz, window)

if __name__ == "__main__":
    window.mainloop()

```

Soluzione 3: In questa terza soluzione bisognerà visualizzare un grafico che ha sull'asse delle X il tempo e sull'asse delle Y i valori del sensore misurati, utilizzando la libreria matplotlib.

codice python guiSensore3: