# Documentazione tecnica: programmazione posizionatore di sonda

import serial

import time // inclusione librerie necessarie

# coordinates in um

x\_um, y\_um, z\_um = 10000, 10000, 10000 // variabili coordinate in micrometri

porta\_seriale = 'COM3' // porta seriale di collegamento

# conversion from um to mm

def calculate\_mm(micrometri):

return micrometri / 1000 # CONVERSIONE: 1000 µm = 1 mm

# Comando per tornare a home

def return\_Home(assi="X Y Z"):

return f"G28 {assi}"

#alternative -> return f"G28 X0.0 Y0.0 X0.0"

# generation of GCode commands to go to the target point

def generate\_gcode(x\_um, y\_um, z\_um):

x = calculate\_mm(x\_um)

y = calculate\_mm(y\_um)

z = calculate\_mm(z\_um)

gcode = []

gcode.append("M17") #activate stepper motors

gcode.append("G90") #G90 absolute positioning

gcode.append("G21") #set millimeters as unit

gcode.append(return\_Home())

gcode.append(f"G1 X{x:.2f} Y{y:.2f} Z{z:.2f} F3000") #moves linearly to the target point

gcode.append(return\_Home())

gcode.append("M2") #end of program

return gcode

# serial communication

try:

ctr\_connection = serial.Serial(porta\_seriale, 115200, timeout=1) // passiamo porta seriale, velocita bit al sec , timeout tempo

gcode = generate\_gcode(x\_um, y\_um, z\_um)

for line in gcode:

ctr\_connection.write((line+ '\n').encode("ascii")) #encoding to ascii

time.sleep(0.2)

#risposta = ctr\_connection.readline().decode().strip()

print("connessione stabilita!")

#print(f"Risposta: {risposta}")

#printing GCode's commands preveously generated

ctr\_connection.close()

except Exception as e:

print(f"Connessione non riuscita: {e}")