**ARDUINO IDE CODE**

//Initial Constant

const int redLED1 = 6;

const int redLED2 = 5;

const int yellowLED1 = 8;

const int yellowLED2 = 9;

const int greenLED1 = 11;

const int greenLED2 = 12;

//CHecking Command in UI

void setup() {

pinMode(redLED1, OUTPUT);

pinMode(redLED2, OUTPUT);

pinMode(yellowLED1, OUTPUT);

pinMode(yellowLED2, OUTPUT);

pinMode(greenLED1, OUTPUT);

pinMode(greenLED2, OUTPUT);

Serial.begin(9600);

Serial.println("Setup complete");

}

//FeedbCK Command in UI

void loop() {

if (Serial.available() > 0) {

String command = Serial.readStringUntil('\n');

Serial.print("Received command: ");

Serial.println(command);

if (command == "RED") {

Serial.println("Turning on RED LEDs");

digitalWrite(redLED1, HIGH);

digitalWrite(redLED2, HIGH);

delay(2000); // Keep LEDs on for 2 seconds

digitalWrite(redLED1, LOW);

digitalWrite(redLED2, LOW);

Serial.println("Turning off RED LEDs");

} else if (command == "YELLOW") {

Serial.println("Turning on YELLOW LEDs");

digitalWrite(yellowLED1, HIGH);

digitalWrite(yellowLED2, HIGH);

delay(2000); // Keep LEDs on for 2 seconds

digitalWrite(yellowLED1, LOW);

digitalWrite(yellowLED2, LOW);

Serial.println("Turning off YELLOW LEDs");

} else if (command == "GREEN") {

Serial.println("Turning on GREEN LEDs");

digitalWrite(greenLED1, HIGH);

digitalWrite(greenLED2, HIGH);

delay(2000); // Keep LEDs on for 2 seconds

digitalWrite(greenLED1, LOW);

digitalWrite(greenLED2, LOW);

Serial.println("Turning off GREEN LEDs");

} else if (command == "MIX") {

Serial.println("Executing MIX command");

int leds[] = {redLED1, redLED2, yellowLED1, yellowLED2, greenLED1, greenLED2};

for (int i = 0; i < 5; i++) {

int randomLED = leds[random(0, 6)];

Serial.print("Blinking LED on pin ");

Serial.println(randomLED);

digitalWrite(randomLED, HIGH);

delay(500);

digitalWrite(randomLED, LOW);

delay(500);

}

}

// Turn off all LEDs before reading the next command

digitalWrite(redLED1, LOW);

digitalWrite(redLED2, LOW);

digitalWrite(yellowLED1, LOW);

digitalWrite(yellowLED2, LOW);

digitalWrite(greenLED1, LOW);

digitalWrite(greenLED2, LOW);

}

}

**PYCHARM CODE**

**import threading**

**import time**

**import tkinter as tk**

**from PIL import Image, ImageTk**

**import serial**

**mySerial = serial.Serial('COM4', 9600, timeout=1)**

**# MESSAGES ACTION**

**def send\_command(*command*, *button*):**

**mySerial.write((*command* + '\n').encode('utf-8'))**

**update\_button\_status(*button*)**

**update\_display\_textbox(f'Sent command: {*command*}')**

**def clear\_message\_action():**

**display\_textbox.config(state=tk.NORMAL)**

**display\_textbox.delete(1.0, tk.END)**

**display\_textbox.config(state=tk.DISABLED)**

**def update\_button\_status(*button*):**

**# Reset all buttons to their original colors**

**for b in buttons:**

**b.config(bg=original\_colors[b])**

**# Highlight the pressed button**

***button*.config(bg="gray")**

**# DEF LISTEN SERIAL**

**def listen\_serial():**

**while True:**

**if mySerial.in\_waiting > 0:**

**dataReceived = mySerial.readline().decode('utf-8').strip()**

**print(dataReceived)**

**update\_display\_textbox(dataReceived)**

**time.sleep(0.1)**

**def update\_display\_textbox(*data*):**

**display\_textbox.config(state=tk.NORMAL)**

**display\_textbox.insert(tk.END, *data* + '\n')**

**display\_textbox.config(state=tk.DISABLED)**

**root = tk.Tk()**

**root.title("Project Virtuso: DIGICOM in Pydroid")**

**root.geometry("450x650") # Adjusted window size to accommodate the textbox**

**# Create a canvas to hold the GIF image**

**canvas = tk.Canvas(root, width=450, height=650)**

**canvas.pack(fill="both", expand=True) # Make the canvas expand to fill the window**

**root.mainloop()**

**## BACKGROUND OF USER INTERFACES ##**

**# Load the GIF image**

**img = Image.open("7d1mr75y0r471.gif")**

**frames = []**

**for i in range(img.n\_frames):**

**img.seek(i)**

**frame = ImageTk.PhotoImage(img)**

**frames.append(frame)**

**# Create a label with the GIF image**

**gif\_label = tk.Label(root, image=frames[0])**

**gif\_label.image = frames[0]**

**gif\_label.place(relx=0.5, rely=0.5, anchor=tk.CENTER)**

**# Update the GIF image every few milliseconds**

**def update\_gif(*frame*):**

**gif\_label.config(image=frames[*frame*])**

**root.after(50, lambda: update\_gif((*frame* + 1) % len(frames)))**

**update\_gif(0)**

**# Title Label**

**title\_label = tk.Label(root, text="WELCOME TO PROJECT VIRTOUSO", font=("Courier", 17, "bold italic"), bg="#000000", fg="#FFFFFF", borderwidth=5, relief="ridge")**

**title\_label.place(relx=0.5, rely=0.05, anchor=tk.CENTER)**

**# Subtitle Label**

**subtitle\_label = tk.Label(root, text="Click some button to turn on\n each specific Arduino LED", font=("Courier", 12, "italic"), bg="#000000", fg="#FFFFFF")**

**subtitle\_label.place(relx=0.5, rely=0.15, anchor=tk.CENTER)**

**# Add buttons for each command with status indicator**

**original\_colors = {}**

**buttons = []**

**# Buttons for Selections with improved design**

**button\_style = {"font": ("Courier", 12, "bold"), "height": 1, "width": 8, "borderwidth": 3, "relief": "raised"}**

**red\_button = tk.Button(root, text="RED", bg="#FF0000", fg="#FFFFFF", command=lambda: send\_command("RED", red\_button), \*\*button\_style)**

**red\_button.place(relx=0.3, rely=0.3, anchor=tk.CENTER)**

**original\_colors[red\_button] = "#FF0000"**

**buttons.append(red\_button)**

**yellow\_button = tk.Button(root, text="YELLOW", bg="#FFFF00", fg="#000000", command=lambda: send\_command("YELLOW", yellow\_button), \*\*button\_style)**

**yellow\_button.place(relx=0.7, rely=0.3, anchor=tk.CENTER)**

**original\_colors[yellow\_button] = "#FFFF00"**

**buttons.append(yellow\_button)**

**green\_button = tk.Button(root, text="GREEN", bg="#00FF00", fg="#FFFFFF", command=lambda: send\_command("GREEN", green\_button), \*\*button\_style)**

**green\_button.place(relx=0.3, rely=0.4, anchor=tk.CENTER)**

**original\_colors[green\_button] = "#00FF00"**

**buttons.append(green\_button)**

**mix\_button = tk.Button(root, text="MIX", bg="#800080", fg="#FFFFFF", command=lambda: send\_command("MIX", mix\_button), \*\*button\_style)**

**mix\_button.place(relx=0.7, rely=0.4, anchor=tk.CENTER)**

**original\_colors[mix\_button] = "#800080"**

**buttons.append(mix\_button)**

**clear\_button = tk.Button(root, text="CLEAR", bg="#000000", fg="#FFFFFF", command=clear\_message\_action, \*\*button\_style)**

**clear\_button.place(relx=0.5, rely=0.5, anchor=tk.CENTER)**

**# Textbox for displaying messages with a border**

**display\_textbox = tk.Text(root, width=50, height=5, state=tk.DISABLED, borderwidth=3, relief="ridge")display\_textbox.pl**

**ace(relx=0.5, rely=0.8, anchor=tk.CENTER)**

**# Add some space below the buttons**

**space\_label = tk.Label(root, text="", height=2, bg="#000000")**

**space\_label.place(relx=0.5, rely=0.7, anchor=tk.CENTER)**

**# SerialThreading**

**serial\_thread = threading.Thread(target=listen\_serial, daemon=True)**

**serial\_thread.start()**

**root.mainloop()**