#!/usr/bin/python3

from signal import signal, SIGTERM, SIGHUP, pause

from smbus import SMBus

from gpiozero import PWMLED

from time import sleep

import sys

bus = SMBus(1)

ads7830\_commands = (0x84, 0xc4, 0x94, 0xd4, 0xa4, 0xe4, 0xb4, 0xf4)

top = PWMLED(5) # GPIO 5

right = PWMLED(20)

bottom = PWMLED(26)

left = PWMLED(13)

led\_19 = PWMLED(19) # GPIO 19

led\_27 = PWMLED(27) # GPIO 27

def safe\_exit(signum, frame):

exit(1)

def read\_ads7830(input):

bus.write\_byte(0x4b, ads7830\_commands[input])

return bus.read\_byte(0x4b)

def no\_drift(input):

value = read\_ads7830(input)

return value if value < 110 or value > 140 else 127

def read\_min(input):

while True:

value = read\_ads7830(input)

yield (127 - value) / 127 if value < 110 else 0

def read\_max(input):

while True:

value = read\_ads7830(input)

yield (value - 128) / 127 if value > 140 else 0

# Display initial message

print("Your lock is active.")

# Function to prompt user after top GPIO is lit

def prompt\_for\_gpio\_blink():

print("Press 'y' to unlock the lock or press 'n'to display your access denied! ")

try:

signal(SIGTERM, safe\_exit)

signal(SIGHUP, safe\_exit)

# Monitor joystick movements and light up LEDs accordingly

top.source = read\_max(6) # Monitor the top direction

right.source = read\_max(7)

bottom.source = read\_min(6)

left.source = read\_min(7)

while True:

value = read\_ads7830(6) # Read the joystick for the top direction

if value < 110:

top.value = 1 # Top direction lit

print("password is correct.")

prompt\_for\_gpio\_blink()

# Wait for user input to blink one of the LEDs

user\_input = input("Enter 'y' or 'n': ").strip().lower()

if user\_input == 'y':

print("Access Granted!")

led\_19.blink(on\_time=0.5, off\_time=0.5)

sleep(5) # Blink for 5 seconds

led\_19.off() # Turn off after blinking

elif user\_input == 'n':

print("Access Denied!")

led\_27.blink(on\_time=0.5, off\_time=0.5)

sleep(5) # Blink for 5 seconds

led\_27.off() # Turn off after blinking

else:

print("Invalid input. Please press 'y' or 'n'.")

# Other joystick movements can be handled here if needed

sleep(0.1)

except KeyboardInterrupt:

pass

finally:

# Cleanup

top.source = None

right.source = None

bottom.source = None

left.source = None

top.close()

right.close()

bottom.close()

left.close()

led\_19.close()

led\_27.close()