# Import libraries

import time

import serial # For serial communication with sensors

import matplotlib.pyplot as plt # For data visualization

# Define constants (modify these based on your setup)

PORT = "COM3" # Adjust for your serial port

BAUDRATE = 9600 # Adjust for your sensor baud rate

# Sensor data format (modify based on sensor output format)

VOLTAGE\_OFFSET = 0 # Offset value for voltage reading

CURRENT\_OFFSET = 0 # Offset value for current reading

# Function to read sensor data

def read\_sensor\_data():

ser = serial.Serial(PORT, BAUDRATE)

data = ser.readline().decode("utf-8").strip().split(",") # Read data, decode, split

voltage = float(data[0]) + VOLTAGE\_OFFSET # Extract voltage with offset

current = float(data[1]) + CURRENT\_OFFSET # Extract current with offset

ser.close()

return voltage, current

# Function to calculate power

def calculate\_power(voltage, current):

power = voltage \* current

return power

# Function to store data (replace with your preferred data storage method)

def store\_data(timestamp, voltage, current, power):

# Implement logic to store data in a database, CSV file, etc.

print(f"Timestamp: {timestamp}, Voltage: {voltage}V, Current: {current}A, Power: {power}W")

# Example: data.append([timestamp, voltage, current, power])

# Main loop for continuous monitoring

data = [] # List to store data (modify based on storage function)

while True:

timestamp = time.time() # Capture timestamp

voltage, current = read\_sensor\_data()

power = calculate\_power(voltage, current)

store\_data(timestamp, voltage, current, power)

data.append([timestamp, voltage, current, power]) # Example data storage

# Add logic to control data collection frequency (e.g., every minute)

time.sleep(60) # Replace with desired delay between readings

# Data analysis and visualization (after data collection)

# Example: Plot power vs time

timestamps, power\_values = zip(\*data) # Unpack data for plotting

plt.plot(timestamps, power\_values)

plt.xlabel("Time (seconds)")

plt.ylabel("Power (Watts)")

plt.title("Solar Panel Power Output")

plt.show()