This script is designed to subscribe to an MQTT topic, receive messages, and store them in a local Excel (.xls) file. Below is a detailed breakdown of the script:

### ****1. Importing Required Libraries****

python

CopyEdit

import paho.mqtt.client as mqtt

import xlwt

import xlrd

from datetime import datetime

import os

* paho.mqtt.client: A Python MQTT client library used to connect and communicate with an MQTT broker.
* xlwt: A library to create and write .xls (Excel 97-2003 format) files.
* xlrd: A library to read existing .xls files.
* datetime: Provides date and time functions to timestamp received messages.
* os: Used to check if an Excel file already exists.

### ****2. Configuration Variables****

python

CopyEdit

MQTT\_BROKER\_IP = "your.broker.ip" # MQTT broker IP address

MQTT\_PORT = 1883 # MQTT broker port

MQTT\_TOPIC = "your/data/topic" # Topic to subscribe to

EXCEL\_FILE = "received\_data.xls" # Output Excel file name

* These variables store MQTT broker details (IP address, port) and the topic name.
* The script will subscribe to the given MQTT topic and save received data in received\_data.xls.

### ****3. Checking for Existing Excel File****

python

CopyEdit

existing\_data = []

# Initialize or load existing Excel data

if os.path.exists(EXCEL\_FILE):

try:

workbook = xlrd.open\_workbook(EXCEL\_FILE)

sheet = workbook.sheet\_by\_index(0)

for row\_idx in range(sheet.nrows):

existing\_data.append(sheet.row\_values(row\_idx))

except Exception as e:

print(f"Error reading existing file: {e}")

existing\_data = [["Timestamp", "Message"]]

else:

existing\_data = [["Timestamp", "Message"]]

* Checks if the file received\_data.xls exists.
* If the file exists:
  + It reads the content using xlrd and stores it in existing\_data.
  + If there's an error (e.g., file corruption), it resets existing\_data with a default header (["Timestamp", "Message"]).
* If the file does **not** exist, it initializes existing\_data with the same header.

### ****4. Function to Write Data to Excel****

python

CopyEdit

def write\_to\_excel(data):

"""Writes all data to the Excel file"""

try:

workbook = xlwt.Workbook()

sheet = workbook.add\_sheet("MQTT Data")

# Write all rows

for row\_idx, row in enumerate(data):

for col\_idx, value in enumerate(row):

sheet.write(row\_idx, col\_idx, value)

workbook.save(EXCEL\_FILE)

print(f"Data successfully written to {EXCEL\_FILE}")

except Exception as e:

print(f"Error writing to Excel: {e}")

* Creates a new Excel workbook (.xls file).
* Writes all stored data (existing\_data) into an Excel sheet named "MQTT Data".
* If an error occurs, it prints an error message.

### ****5. MQTT Callback Functions****

#### ****5.1.**** on\_connect

python

CopyEdit

def on\_connect(client, userdata, flags, rc):

if rc == 0:

print("Connected to MQTT Broker!")

client.subscribe(MQTT\_TOPIC)

else:

print(f"Connection failed with code {rc}")

* Called when the client successfully connects to the MQTT broker.
* If rc == 0, it means the connection was successful, and the client subscribes to the given topic.
* Otherwise, it prints an error message.

#### ****5.2.**** on\_message

python

CopyEdit

def on\_message(client, userdata, msg):

try:

timestamp = datetime.now().strftime('%Y-%m-%d %H:%M:%S')

payload = msg.payload.decode("utf-8")

print(f"Received message: {payload}")

# Add new entry to data

existing\_data.append([timestamp, payload])

# Update Excel file

write\_to\_excel(existing\_data)

except Exception as e:

print(f"Error processing message: {e}")

* Triggered when an MQTT message is received.
* Extracts the timestamp and message payload.
* Appends the new message to existing\_data.
* Calls write\_to\_excel() to update the Excel file.
* If an error occurs, it prints an error message.

### ****6. Setting Up and Connecting the MQTT Client****

python

CopyEdit

client = mqtt.Client()

client.on\_connect = on\_connect

client.on\_message = on\_message

* Creates an MQTT client instance.
* Assigns on\_connect and on\_message functions to handle connection and messages.

### ****7. Connecting to MQTT Broker****

python

CopyEdit

try:

client.connect(MQTT\_BROKER\_IP, MQTT\_PORT, 60)

except Exception as e:

print(f"Connection error: {e}")

exit(1)

* Attempts to connect to the MQTT broker.
* If an error occurs, it prints an error message and exits.

### ****8. Listening for Messages****

python

CopyEdit

print(f"Listening for messages on topic '{MQTT\_TOPIC}'...")

client.loop\_forever()

* Prints a message indicating that the script is listening for messages.
* loop\_forever() keeps the script running to continuously receive messages.