UDP CAN client and server

Introduction

In this task you will be asked to create CAN frame UDP client - server script, the purpose of this task is to evaluate your understanding of:

- Development in Python language.
- Network understanding.
- Ability to learn new things.

Basic rules

- Use python 2.7
- Don’t cheat
- Document your code
- Add an example execution output
- Send all data as zip or a link to git repository (preferred)

Task:

1. Write a client class that can open a csv file containing a list of CAN frames and send them one by one to a UDP server for a given address and port.
    a. When the client finished it will exit cleanly.
2. Write a server class that listen to UDP address and port to CAN frames (please see CAN packet representation section)
    a. The server will parse the received data.
    b. The server will create the following statistics:
        i. How many times a specific CAN msg-id appear.
        ii. What is the sum of all messages data length received by the server.
        iii. How many packets received by the server.
    c. The server will write the statistics to log files.
    d. The server will write all frames to a csv file in the same format as attached.
    e. The server will stop after a specified timeout (socket timeout) and flush all data to log and output files.
3. Create a main method that handle arguments and run the server and the client.

The script receive the following arguments:

```
--client - If set the script will create a client object
--server - If set the script will create a server object
--port - port to send or listen to
--address - address to send or listen to
--CSV-file-path - path to csv file to read or write to.
--timeout - timeout for the server in seconds

argus_udp_script.py --client --address 127.0.0.1 --port 5000 --csv-file-path can_frames.csv
argus_udp_script.py --server --address 127.0.0.1 --port 5000 --csv-file-path res_frames.csv
--timeout 30
```

The client will send to the port and address provided by the user CAN frames loaded from provided csv file (Comma separated values).

CAN packet representation

Name	Type	Size	Comments
MSG_ID	Bytes	4	A number between 0x0 and 0xFFFFFFFF
MSG_LENGTH	Bytes	1	Number between 0 and 8
Data	Bytes	8	List of bytes (0x0 to 0xFF)

e.g CAN packet

MSG_ID	LENGTH	DATA
FF	5	AABBCCDDEE

Will be sent as a buffer of FF05AABBCCDDEE000000 and will receive as FF05AABBCCDDEE

References

Socket - https://docs.python.org/2/library/socket.html
Csv file - https://docs.python.org/2/library/csv.html
Struct (for packing bytes) - https://docs.python.org/2/library/struct.html
CAN - https://en.wikipedia.org/wiki/CAN_bus
Hexadecimal - https://en.wikipedia.org/wiki/Hexadecimal
PEP-8 - https://www.python.org/dev/peps/pep-0008