Option II

For Option II, you must choose a task with simple SQL. Different tasks (1.1 and 1.2) are necessary so that you do not repeat your selfwork.

1. SQL

You will need:

- MS Excel (or Numbers, or Libre Calc, or notepad)
- MySQL, PostgreSQL or other relational database management system

The "data" folder contains csv good files (data0 ... data9) and csv “data_bad”. You must choose data_bad and 3 files from data0…data9.

You need to develop a database structure (the database must contain at least 3 tables). Load data from the csv files of your choice into the database. You have to figure out what to do with the bad file, how to normalize it, and how to use it later. In order to understand what the problem is with a bad file, you need to use good files.

TASKS SQL

You also need to write the following SQL queries (choose any 3 from task 1.1 and any 3 from task 1.2:

Task 1.1

1. Count the number of unique Locations
2. Find all Customers whose LightSen is in the range between 4500-7000
3. Find all Customer that contains letters: LC and NC. (the letters go in this order, but between them there may be other letters and symbols. Letters are not case-sensitive)

4. Find the Location with the largest number of Customer, while id is ignored after the name. Customer id is any number after name or the symbol ("#" , ":" , "/" or other)
5. Using one SELECT, display the following parameters: the maximum range of the TempSen parameter (TempSenMax-TempSenMin) for all Customers; the average LightSen value for board STM, the number of Customers that iosApp support
6. Display a list of all unique customers
7. Find the Location with the maximum total time (allTimeWork) of all Customers that belong to it
8. Find all Customers with GyrSen in the range between 22-24
9. Find the Customer with the maximum HumpSen value, if there are more than one such Customer, then display who has the highest allTimeWork

Task 1.2

1. Move all identifiers from customer and location to the corresponding tables and remove their mention in the name
2. Find all iosapp Customers that do not support wi-fi and update them to “yes”
3. Find all iosapp Customers that do not support wi-fi and update them to “yes”
4. Create a view object that will display upon request: the number of CNCs, the total operating time of all CNCs, the number of iot sensors, the total operating time iot sensors
5. Find all Customers with a Number of sensor in the range between 25-27, and for all devices with STM or ESP, regardless of digits further instead of Yes in the iOSApp column display "1", and for No - "0"
6. Create a table price, which will indicate the cost of devices: for PLC - set 1000, CNC-10000, iot-100 for all the others arbitrarily
7. Calculate the total cost of all CNC, PLC, IOT and other devices

2. Analysis data

You will need: python or another programming language, maybe web.

You must analyze the data received from the motor to detect incorrect operation.

The “drive” folder contains xml files from 1 to 5. 3 of these files of good work and 2 of bad work.

You need find 3 good and 2 bad files.

Parameters in files (4 channel of oscilloscope):

S-0-0047.0.0 Position command value

S-0-0051.0.0 Position feedback value

S-0-0040.0.0 Velocity feedback value

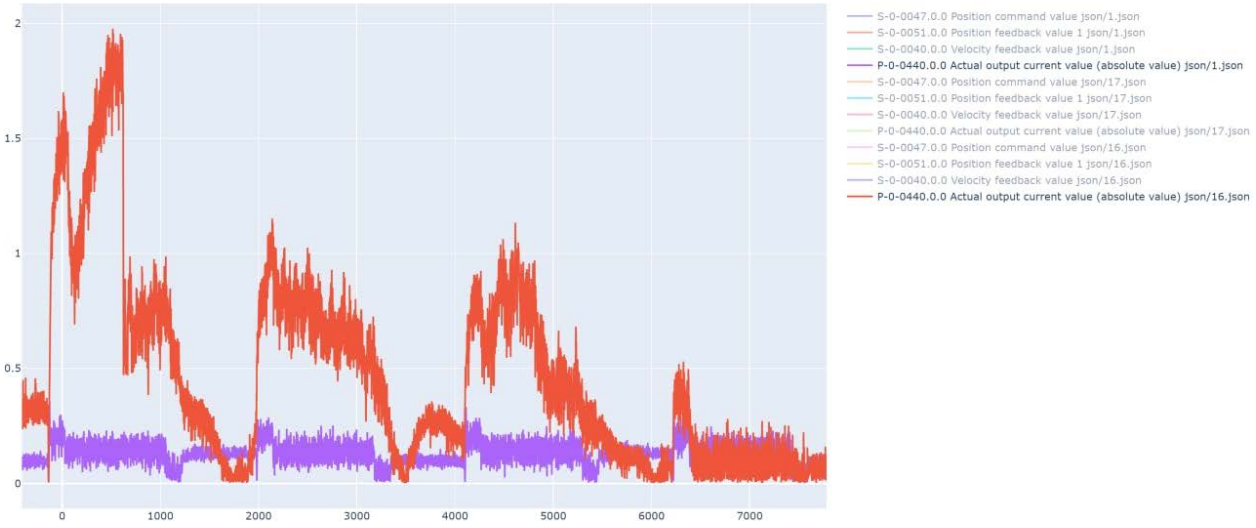
P-0-0440.0.0 Actual output current value (absolute value)

Data on X – time ms, Data Y – value of parameter.

TASKS ANALYSIS

You can use simple visualization for analyze, but it's better if you can write a script that can analyze the data and find problems. Some measurement channels may show approximately the same result for all files. You need to find channels through which to find problems and find this problem.

Signals


2
1.5
1
0.5
0
0
1000
2000
3000
4000
5000
6000
7000
S-0-0047.0.0 Position command value json/1.json
S-0-0051.0.0 Position feedback value 1 json/1.json
S-0-0040.0.0 Velocity feedback value json/1.json
P-0-0440.0.0 Actual output current value (absolute value) json/1.json
S-0-0047.0.0 Position command value json/17.json
S-0-0051.0.0 Position feedback value 1 json/17.json
S-0-0040.0.0 Velocity feedback value json/17.json
P-0-0440.0.0 Actual output current value (absolute value) json/17.json
S-0-0047.0.0 Position command value json/16.json
S-0-0051.0.0 Position feedback value 1 json/16.json
S-0-0040.0.0 Velocity feedback value json/16.json
P-0-0440.0.0 Actual output current value (absolute value) json/16.json