Exercise – 4

Published on: 06.05.2024

Deadline: 13.05.2024 – 1:59pm

Submission Instructions:

- Clone the Exercise GitHub repository: https://github.com/ETCE-LAB/ETCE-Exercises/
- The exercise is given to you in the form of a jupyter notebook. You can either install jupyter on your personal system, or use https://jupyter-cloud.gwdg.de/ (TIP: Jupyter Cloud allows you to clone the exercise repositorso you don't have to upload the jupyter notebook and other files manually)
- Programming language: Python 3.10+
  - **You only need to modify the „solution.ipynb“ file**. More detailed instructions on where you need to insert your code can be found in this file.
  - Process all available data samples and build up a data structure according to the template given in the file. UUIDs should be unique for each entry. The timestamp field should be filled with an integer corresponding to a UNIX timestamp (UTC) converted from ''yyyy-mm-dd hh:mm:ss.ms CEST'' format.
  - For each timestamp, if data from multiple sources is available for the same field in the data structure, please compute the average.
  - The final cell in the solution.ipynb file grades your solution.
  - This will give you feedback on your solution.
- To submit your solution, upload your modified ‚solution.ipynb‘ file to Moodle.

Task Description – IoT Sensing – Energy sellers

- Imagine a smart electric vehicle (EV) charging station that is connected to a solar panel. The goal of the smart charging station is to serve as many EVs with as much solar energy as possible. To do so, the smart charging station is interested in weather information (e.g., temperature, precipitation, cloudiness) from different sources. Bad (rainy, cloudy) indicates a lower output from the solar panel and thus less energy to charge EVs. However, lots of sun and a blue sky increase the solar panel output which allows the smart charging station to sell more energy.
- Your task is to create a program that queries (senses) the different data sources for weather information.
- The different sources provide different formats of weather data. Therefore, you need to transform it into a single data format for later processing. The solution() function has code snippets that can help you determine what the data from the weather stations look like.
- The expected output of your program is a combined set of weather data (that adheres to the defined format and structure) that contains all of the relevant weather data information from all available sensors.
- Voltage readings are from a TMP36 sensor and needs to be converted to temperature in °C before averaging.

Questions? : etce-etce@tu-clausthal.de