|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных технологий

**Отчет по практическим работам №8**

по дисциплине «Технологические основы Интернета вещей»

|  |  |
| --- | --- |
| **Выполнил:**  Студент группыИМБО-02-22 | Ким Кирилл Сергеевич |
| **Проверил:** | Синицын Иван Васильевич |

2024 г.

# **Практическая работа №8 – Визуализация данных в Интернете вещей**

**Цель работы:**

**Выполнения практической работы**

1. Подпишитесь на несколько MQTT топиков стенда согласно вариантам c компьютера в аудитории или личного устройства и собирайте данные с датчиков в течение 10 или более минут.

Получаемые данные должны сохраняться в локальную базу данных или CSV-файл (на выбор).

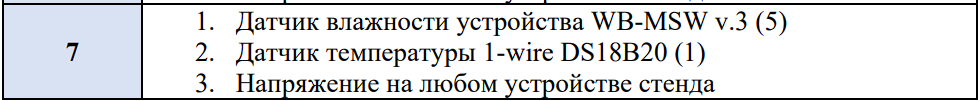


Рисунок 1 — Задания варианта 7

1. Напишите скрипты, позволяющие визуализировать полученные статистические данные в виде:

* Столбиковой диаграммы (гистограммы) (например, частоты показаний датчика)
* Линейного графика (например, показаний по времени)
* Круговой диаграммы (например, распределения показаний по времени)

Программный код для для визуализации полученных статистических данных представлен на листинге 1.

*Листинг 1 - Визуализация*

|  |
| --- |
| from datetime import datetime  import matplotlib.pyplot as plt  import json  # Функция получения данных из json-файла  *def* get\_data\_from\_json(*filename*):      with open(*filename*, 'r') as file:          file\_data = json.load(file)      return file\_data  *def* create\_plots(*plots\_data\_lists*):      # Создание графиков для отрисовки данных      fig, axs = plt.subplots(1, 3, *figsize*=(10,5)) # Получим окно с 1 колонкой и 2 столбцами графиков      # fig - окно, в котором будут отрисовываться графики      # axs содержит в себе список графиков для отрисовки на них значений      # Задание набора точек для отрисовки      # Первый аргумент - список значений по оси X, второй аргумент - по оси Y      axs[0].plot(*plots\_data\_lists*['time'], *plots\_data\_lists*['co2'])        # Задание лейблов для осей и графика      axs[0].set\_xlabel('Time')      axs[0].set\_ylabel('Humidity level')      axs[0].set\_title('Humidity')      # Формирование гистограммы      axs[1].hist(*plots\_data\_lists*['light'])      axs[1].set\_xlabel('Voltage level')      axs[1].set\_ylabel('Count')      axs[1].set\_title('Volt')      labels = ['Низкая','Средняя','Высокая']      sizes = [len([t for t in *plots\_data\_lists*['temp'] if t < 25.97]),               len([t for t in *plots\_data\_lists*['temp'] if 25.97 <= t < 26.1]),               len([t for t in *plots\_data\_lists*['temp'] if t >= 26.1])]      axs[2].pie(sizes, *labels*=labels, *autopct*='%1.1f%%', *startangle*=90)      axs[2].axis('equal')      axs[2].set\_title('Temperature')      return fig, axs  *def* main():      plots\_data\_lists = {          'co2': [],          'light': [],          'time': [],          'sound': [],          'temp': []      }      json\_data = get\_data\_from\_json("data.json")      # Заполнение списков с данными, с преобразованием типов      for json\_dict in json\_data:          plots\_data\_lists['time'].append(datetime.fromisoformat(json\_dict.get('time')))          plots\_data\_lists['co2'].append(float(json\_dict.get('humidity')))          plots\_data\_lists['light'].append(float(json\_dict.get('voltage')))          plots\_data\_lists['temp'].append(float(json\_dict.get('temperature')))      fig, axs = create\_plots(plots\_data\_lists)      plt.show()  if \_\_name\_\_ == "\_\_main\_\_":      main() |

*Листинг 2 – Пример данных*

|  |
| --- |
| [      {  *"time"*: "2024-10-21 18:50:29.672748",  *"number"*: "27",  *"voltage"*: "5.02",  *"humidity"*: "42.9",  *"temperature"*: "23.75"      },      {  *"time"*: "2024-10-21 18:50:35.757803",  *"number"*: "27",  *"voltage"*: "5.02",  *"humidity"*: "41.6",  *"temperature"*: "23.75"      },      {  *"time"*: "2024-10-21 18:50:41.654313",  *"number"*: "27",  *"voltage"*: "5.04",  *"humidity"*: "41.2",  *"temperature"*: "24.1875"      },      {  *"time"*: "2024-10-21 18:50:46.899469",  *"number"*: "27",  *"voltage"*: "5.03",  *"humidity"*: "40.2",  *"temperature"*: "24.75"      },      {  *"time"*: "2024-10-21 18:50:52.034503",  *"number"*: "27",  *"voltage"*: "5.02",  *"humidity"*: "39.9",  *"temperature"*: "25.0625"      }, |

A close-up of a graph

Description automatically generated **Рисунок 8 – Визуализация**

**Выводы:**

В процессе выполнения практической работы построены диаграммы, отражающие последовательность записи из топиков из практической работы согласно варианту.