|  |  |  |
| --- | --- | --- |
|  |  |  |
| 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 практической работе №8**

по дисциплине «Технологические основы Интернета вещей»

|  |  |
| --- | --- |
| **Выполнил:**  Студенты группыИВБО-01-22 |  |
| **Проверил:** | Синицын Иван Васильевич. |

2024 г.

СОДЕРЖАНИЕ

[1 ПРАКТИЧЕСКАЯ РАБОТА №8 – ВИРТУАЛИЗАЦИЯ ДАННЫХ В ИНТЕРНЕТЕ ВЕЩЕЙ 3](#_Toc182230671)

[1.1 задание на практическую работу 3](#_Toc182230672)

[1.2 Код программы 3](#_Toc182230673)

[4.3 Описание работы программы 5](#_Toc182230674)

[ВЫВОД 6](#_Toc182230675)

# 1 ПРАКТИЧЕСКАЯ РАБОТА №8 – ВИРТУАЛИЗАЦИЯ ДАННЫХ В ИНТЕРНЕТЕ ВЕЩЕЙ

## 1.1 задание на практическую работу

* + 1. Подпишитесь на несколько MQTT топиков стенда согласно вариантам c компьютера в аудитории или личного устройства и собирайте данные с датчиков в определенного времени (2 вариант):
    - Датчик движения устройства WB-MSW v.3 (5);
    - Датчик температуры устройства WB-MS v.2 (12).
    1. Напишите скрипты, позволяющие визуализировать полученные статистические данные в виде графика значений.

## 1.2 Код программы

Для реализации данные работы были использованы методы для чтения и обработки сигналов по MQTT из ПР №7, удалены методы для обработки XML и добавлен метод при парсинге JSON для создания графика значений.

Код программы приведен в Листинге 4.1.

Листинг 4.1 – Программа обработки сигналов по MQTT и их графический вывод

|  |
| --- |
| import xml.etree.ElementTree as ET  import paho.mqtt.client as mqtt  import json  from datetime import datetime, time  import pandas as pd  import matplotlib.pyplot as plt  import numpy as np  HOST = "192.168.2.22"  PORT = 1883  KEEPALIVE = 60  SUB\_TOPICS = {  '/devices/wb-msw-v3\_21/controls/Sound Level': 'Sound',  '/devices/wb-ms\_11/controls/Illuminance': 'light'  }  JSON\_LIST = []  JSON\_DICT = {}  for value in SUB\_TOPICS.values(): |

Продолжение Листинга 4.1

|  |
| --- |
| JSON\_DICT[value] = 0  def on\_connect(client, userdata, flags, rc):  print("Connected with result code " + str(rc))  for topic in SUB\_TOPICS.keys():  client.subscribe(topic)  def on\_message(client, userdata, msg):  payload = msg.payload.decode()  topic = msg.  param\_name = SUB\_TOPICS[topic]  JSON\_DICT[param\_name] = payload  JSON\_DICT['time'] = str(datetime.now())  JSON\_LIST.append(JSON\_DICT.copy())  print(topic + " " + payload)  with open('data.json', 'w') as file:  json\_string = json.dumps(JSON\_LIST)  file.write(json\_string)  parse\_json('data.json')  n = {"sound": [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],  "light": [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]}  def parse\_json(file\_path):  with open(file\_path, 'r') as json\_file:  data = json.load(json\_file)  print(data)  n["sound"] = n["sound"][1:] + [int(float(data[-1]["Sound"]))]  n["light"] = n["light"][1:] + [int(float(data[-1]["light"]))]  df = pd.DataFrame(n)  x = np.arange(10)  plt.axis([0, 10, 0, 400])  plt.legend(n, loc=2)  plt.plot(x, df)  plt.show()  def main():  client = mqtt.Client()  client.on\_connect = on\_connect  client.on\_message = on\_message  client.connect(HOST, PORT, KEEPALIVE)  client.loop\_forever()  if \_\_name\_\_ == "\_\_main\_\_":  main() |

# ВЫВОД

В данных практических работах были более подробны изучены датчики и измерительные устройства, так же возможности взаимодействия с устройством через командую строку ПК по протоколу SSH.Создание сценариев для мониторинга данных на языке Python, используя протокол MQTT. Сценарии выполнялись при взаимодействии со стендом как физически, так и через веб-интерфейс.