|  |  |  |
| --- | --- | --- |
|  |  |  |
| 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 №5-8**

по дисциплине «Технологические основы Интернета вещей»

|  |  |
| --- | --- |
| **Выполнили:**  Студенты группыИКБО-15-22 | Оганнисян Григор Амбарцумович  Кудинов Артём Вячеславович |
| **Проверил:** | Куликова Ирина Викторовна. |

2024 г.

**СОДЕРЖАНИЕ**

[Практическая работа №5 – Измерительные и исполнительные устройства в Интернете вещей 4](#_Toc181899315)

[Цель работы: 4](#_Toc181899316)

[Часть 1. Измерительные и исполнительные устройства стенда 4](#_Toc181899317)

[Часть 2. Протоколы работы с устройствами 7](#_Toc181899318)

[Практическая работа №6 – Основы работы с протоколом MQTT. Брокераж сообщений. 12](#_Toc181899319)

[Цель работы: 12](#_Toc181899320)

[Часть 1. SSH-подключение 12](#_Toc181899321)

[Часть 2. Подписка на топики 13](#_Toc181899322)

[Часть 3. Управление устройствами 13](#_Toc181899323)

[Практическая работа №7 – Форматы представления данных. 14](#_Toc181899324)

[Цель работы: 14](#_Toc181899325)

[Часть 1. Подписка на топики 14](#_Toc181899326)

[Часть 2. Основной скрипт 14](#_Toc181899327)

[Часть 3. Создание парсера 16](#_Toc181899328)

[Практическая работа №8 – Визуализация данных в Интернете вещей. 18](#_Toc181899329)

[Цель работы: 18](#_Toc181899330)

[Часть 1. Подписка на топики 18](#_Toc181899331)

[Часть 2. Основной скрипт 18](#_Toc181899332)

[Часть 3. Скрипты для получения диаграмм 19](#_Toc181899333)

[ДОПОЛНИТЕЛЬНЫЕ ЗАДАНИЯ 22](#_Toc181899334)

[Дополнительное задание по практической работе № 5 22](#_Toc181899335)

[Часть 1. Оборудование для сборки аппаратных компонентов системы климат-контроля умного дома 22](#_Toc181899336)

[Часть 2. Технологии передачи данных от физического устройства в Интернет 27](#_Toc181899337)

[Обоснование выбора: 27](#_Toc181899338)

[Сравнение с другими технологиями: 28](#_Toc181899339)

[Дополнительное задание по практической работе № 6 30](#_Toc181899340)

[Часть 1. Выбор облачной платформы для системы климат-контроля умного дома 30](#_Toc181899341)

[Часть 2. Построение диаграммы последовательности 32](#_Toc181899342)

[ВЫВОД 33](#_Toc181899343)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 34](#_Toc181899344)

**Практическая работа №5 – Измерительные и исполнительные устройства в Интернете вещей**

**Цель работы:**

Ознакомиться с устройствами в составе стенда и их характеристиками. Описать датчики согласно варианту.

**Вариант № 6:**

1. Модуль обнаружения протечек WB-MWAC (10);
2. Качество воздуха VOC в составе устройства WB-MSW v.3 (5);
3. Датчик протечки (23).

**Часть 1. Измерительные и исполнительные устройства стенда**

**1. Модуль обнаружения протечек WB-MWAC (Water Leak Controller)**

1. Название датчика/устройства:

WB-MWAC (Модуль обнаружения протечек)

2. Тип измерения:

Цифровой (определение наличия или отсутствия протечки)

3. Измеряемые параметры и диапазон измерения:

* Обнаружение протечки на основе контакта сенсора с жидкостью.
* Диапазон измерения: не применяется, так как определяется только наличие/отсутствие протечки.

4. Точность:

Отсутствует, так как модуль предназначен только для детектирования факта протечки.

5. Напряжение питания:

* 9–28В постоянного тока.

6. Уникальный идентификатор датчика в веб-интерфейсе:

* Присваивается индивидуально через веб-интерфейс контроллера, например, wb-mwac-X, где X — уникальный номер устройства.

7. Использующийся протокол передачи данных:

* Modbus RTU

8. Интерфейс управления (шина):

* RS-485

9. Описание входов и выходов, схема подключения:

* Входы:
  + - 4 дискретных входа для подключения сенсоров протечки (сухие контакты).
    - Общие входы для внешних реле/клапанов для отключения подачи воды.
* Выходы:
  + - Реле для управления электромагнитными клапанами (максимальная нагрузка: 250В, 5А переменного тока).
    - Модуль имеет возможность автоматического отключения воды при обнаружении протечки.
* Схема подключения:
  + - Сенсоры подключаются к дискретным входам, электромагнитные клапаны подключаются к релейным выходам. Взаимодействие с контроллером через шину RS-485.

**2. Датчик качества воздуха VOC в составе устройства WB-MSW v.3**

1. Название датчика/устройства:

WB-MSW v.3 (VOC датчик для измерения качества воздуха)

2. Тип измерения:

Аналоговый (измерение концентрации летучих органических соединений – VOC)

3. Измеряемые параметры и диапазон измерения:

* Измерение уровня летучих органических соединений (VOC) в воздухе.
* Диапазон измерения VOC: от 0 pmm до 60000 ppb.

4. Точность:

* Точность измерения: ±15% от показаний.

5. Напряжение питания:

* 9 – 28В постоянного тока.

6. Уникальный идентификатор датчика в веб-интерфейсе:

* Например, wb-msw-v3-X, где X — уникальный номер устройства.

7. Использующийся протокол передачи данных:

* Modbus RTU

8. Интерфейс управления (шина):

* RS-485

9. Описание входов и выходов, схема подключения:

* Входы:
  + - Входы для датчиков температуры, влажности и (при необходимости).
* Выходы:
  + - Цифровой выход для отправки данных о концентрации VOC по шине RS-485.
* Схема подключения:
  + - Подключение устройства к системе автоматизации через интерфейс RS-485. Устройство может одновременно измерять несколько параметров окружающей среды, таких как температура, влажность и , дополнительно к VOC.

**3. Датчик протечки**

1. Название датчика/устройства:

Датчик протечки (например, простейший датчик протечки типа SWS-01)

2. Тип измерения:

Цифровой (наличие/отсутствие протечки воды)

3. Измеряемые параметры и диапазон измерения:

* Обнаружение протечки воды путем определения замыкания контактов при попадании воды.
* Диапазон измерения: фиксируется факт наличия или отсутствия протечки (нет количественного измерения).

4. Точность:

Точность не имеет значения, так как датчик фиксирует только событие – появление протечки.

7. Использующийся протокол передачи данных:

* Сигнал с датчика передается как дискретный сигнал в зависимости от схемы подключения (например, через сухие контакты).

8. Интерфейс управления (шина):

* Подключение напрямую к контроллеру через дискретные входы.

9. Описание входов и выходов, схема подключения:

* Входы:
  + - Вход для сигнала с датчика, подключаемого через замыкающиеся контакты при попадании воды.
* Выходы:
  + - Сухие контакты передают сигнал о факте протечки на контроллер.

**Часть 2. Протоколы работы с устройствами**

**1. Modbus RTU**

Принцип работы:

Modbus RTU (Remote Terminal Unit) – это протокол передачи данных по последовательной линии (RS-232, RS-485). Основой протокола является мастер-слейв (ведущий-ведомый) архитектура, где один мастер инициирует запросы к ведомым устройствам, которые отправляют ответы. Данные передаются в двоичном формате, а для коррекции ошибок используется контрольная сумма CRC (Cyclic Redundancy Check).

Преимущества:

* Простой в реализации и широко поддерживается различными устройствами.
* Эффективная передача данных за счёт использования бинарного формата.
* Возможность работы в сложных промышленных условиях на больших расстояниях по RS-485 (до 1200 метров).

Недостатки:

* Ограниченная пропускная способность (обычно до 115.2 кбит/с).
* Отсутствие стандартизованного управления доступом к шине – только одно устройство может инициировать передачу.
* Низкая гибкость и ограниченные возможности в сравнении с более современными протоколами.

Сфера применения:

* Промышленная автоматизация.
* Управление оборудованием (контроллеры, датчики).
* Системы управления HVAC (Heating, Ventilation, Air Conditioning).

**2. 1-Wire**

Принцип работы:

1-Wire – это однопроводный протокол передачи данных, где один провод используется для передачи данных и питания устройств. Устройства подключаются к шине с помощью общего провода и адресуются по уникальному 64-битному идентификатору. Сигнал управляется хост-устройством, а ведомые устройства могут отвечать по очереди.

Преимущества:

* Простота подключения: требуется всего один провод для передачи данных и питания.
* Низкие затраты на реализацию, что делает его популярным для низкоскоростных приложений.
* Возможность подключения большого числа устройств к одной шине.

Недостатки:

* Ограниченная длина кабеля (до 100 метров).
* Низкая скорость передачи данных (обычно 16.3 кбит/с).
* Чувствительность к шумам и помехам, особенно на длинных линиях связи.

Сфера применения:

* Датчики температуры (например, популярные датчики DS18B20).
* Идентификация и контроль доступа (например, с помощью iButton).
* Применение в системах, где не требуется высокая скорость передачи данных и большие расстояния.

**3. (Inter-Integrated Circuit)**

Принцип работы:

– это синхронный последовательный протокол для передачи данных между микроконтроллерами и периферийными устройствами (датчиками, памятью и т.д.). Он использует две линии – SCL (Serial Clock Line) для тактовых сигналов и SDA (Serial Data Line) для передачи данных. Все устройства подключаются к одной шине и имеют уникальные адреса. Протокол поддерживает режимы мастер и слейв.

Преимущества:

* Простая архитектура с использованием только двух проводов.
* Возможность подключения до 127 устройств на одной шине.
* Широкая поддержка среди микроконтроллеров и периферийных устройств.
* Поддержка как медленных, так и высокоскоростных режимов передачи данных (до 5 Мбит/с в некоторых случаях).

Недостатки:

* Ограниченная длина шины (обычно до 1 метра).
* Низкая скорость по сравнению с другими шинами, такими как SPI.
* Возможны конфликты данных, если несколько устройств одновременно пытаются передавать информацию.

Сфера применения:

* Микроконтроллеры и периферийные устройства (датчики, память, экраны).
* Встроенные системы и электроника.
* Применение в бытовой технике и устройствах умного дома.

**4. CAN (Controller Area Network)**

Принцип работы:

CAN – это многомастерный протокол передачи данных, разработанный для работы в условиях повышенных электромагнитных помех, например, в автомобильной промышленности. Протокол использует приоритетную систему сообщений, передаваемых по двум проводам (CAN\_H и CAN\_L). При конфликте между устройствами, система приоритетов позволяет передать более важные сообщения без задержек.

Преимущества:

* Высокая надёжность и устойчивость к электромагнитным помехам.
* Возможность работы на больших расстояниях и при скорости (до 1 Мбит/с).
* Приоритетная система передачи данных, что делает его идеальным для критичных к времени приложений.

Недостатки:

* Сложность в реализации в сравнении с более простыми протоколами, такими как или UART.
* Ограниченная длина шины на высоких скоростях (до 40 метров при скорости 1 Мбит/с).

Сфера применения:

* Автомобильная промышленность (диагностика, контроль систем ABS, управление двигателем).
* Промышленная автоматизация.
* Робототехника и распределённые системы управления.

**Практическая работа №6 – Основы работы с протоколом MQTT. Брокераж сообщений.**

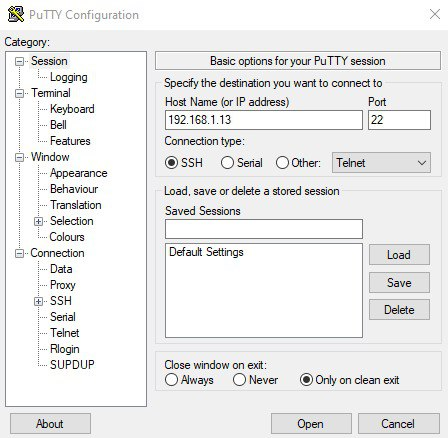
**Цель работы:**

Ознакомиться с подключением к топикам датчиков, а также получать информацию с датчиков. Изучить работы с протоколом MQTT.

**Вариант № 2:**

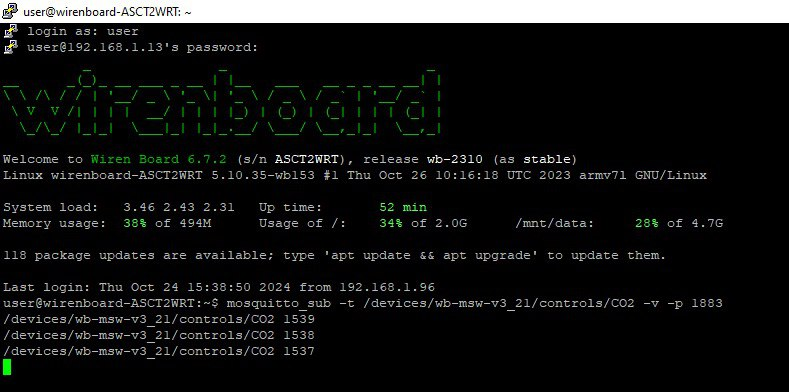
1. Датчик устройства WB-MSW v.3 (5);
2. Датчик протечки (23).

**Часть 1. SSH-подключение**

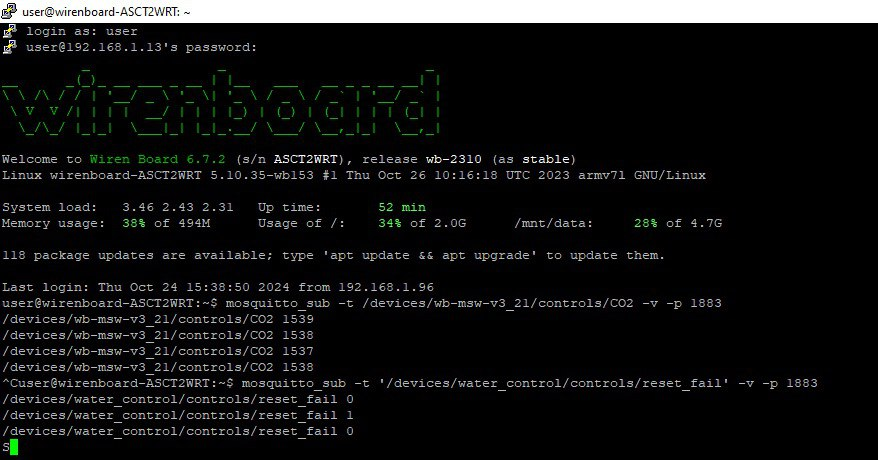


**Рисунок 6.1 – Подключение к устройству**

**Часть 2. Подписка на топики**



**Рисунок 6.2 – Подписка на топик датчика устройства**



**Рисунок 6.3 – Подписка на топик датчика протечки устройства**

**Часть 3. Управление устройствами**



**Рисунок 6.4 – Управление звуковым сигналом**

При выполнении первого запроса издается звуковой сигнал. При выполнении второго запроса звуковой сигнал выключается.



**Рисунок 6.5 – Управление вентилятором**

При выполнении первого запроса включается вентилятор. При выполнении второго запроса вентилятор выключается.

**Практическая работа №7 – Форматы представления данных.**

**Цель работы:**

Разработать программное обеспечение для автоматизированного сбора, упаковки и последующего анализа данных с различных датчиков, подключенных через MQTT-протокол.

**Вариант № 5:**

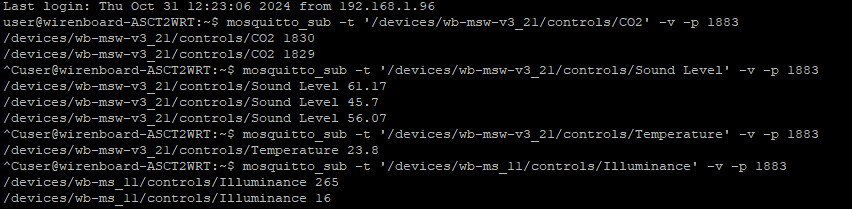
1. Датчик устройства WB-MSW v.3 (5);

2. Датчик шума устройства WB-MSW v.3 (5);

3. Датчик освещенности устройства WB-MS v.2 (12);

4. Датчик температуры устройства WB-MSW v.3 (5).

**Часть 1. Подписка на топики**



**Рисунок 7.1 – Подписка на все топики по варианту**

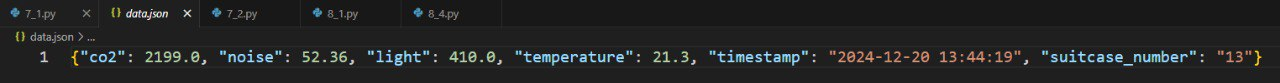
**Часть 2. Основной скрипт**

*Листинг А – Основной скрипт для формирования двух файлов*

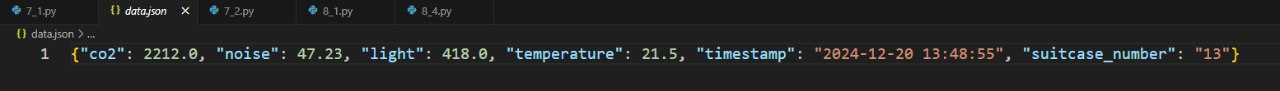
|  |
| --- |
| import json  import xml.etree.ElementTree as ET  import time  import paho.mqtt.client as mqtt  # Глобальные переменные для хранения данных датчиков |

*Продолжение листинга А*

|  |
| --- |
| data = {  "co2": None,  "noise": None,  "light": None,  "temperature": None,  "timestamp": None,  "suitcase\_number": "13" # Замените на последние две цифры вашего IP  }  # Функция обратного вызова при получении сообщения  def on\_message(client, userdata, msg):  topic = msg.topic  value = float(msg.payload.decode())  if "/controls/CO2" in topic:  data["co2"] = value  elif "/controls/Sound Level" in topic:  data["noise"] = value  elif "/controls/Temperature" in topic:  data["temperature"] = value  elif "/controls/Illuminance" in topic:  data["light"] = value  # Настройка MQTT-клиента  client = mqtt.Client()  client.on\_message = on\_message  # Подписка на топики  client.connect("192.168.1.13", 1883, 60) # Замените на адрес вашего MQTT брокера  client.subscribe("/devices/wb-msw-v3\_21/controls/CO2")  client.subscribe("/devices/wb-msw-v3\_21/controls/Sound Level")  client.subscribe("/devices/wb-msw-v3\_21/controls/Temperature")  client.subscribe("/devices/wb-ms\_11/controls/Illuminance")  client.loop\_start()  # Главный цикл для упаковки данных каждые 5 секунд  while True:  data["timestamp"] = time.strftime("%Y-%m-%d %H:%M:%S")  # Упаковка данных в JSON  with open("data.json", "w") as json\_file:  json.dump(data, json\_file)  # Упаковка данных в XML  root = ET.Element("Data")  for key, value in data.items():  element = ET.SubElement(root, key)  element.text = str(value)  tree = ET.ElementTree(root)  tree.write("data.xml")  print("Данные успешно сохранены в файлы JSON и XML.")  time.sleep(5) |



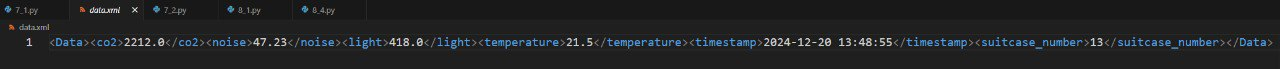
**Рисунок 7.2 – Данные через первые 5 секунд после запуска скрипта (json файл)**



**Рисунок 7.3 – Данные через 10 секунд после запуска скрипта (json файл)**



**Рисунок 7.4 – Данные через первые 5 секунд после запуска скрипта (xml файл)**

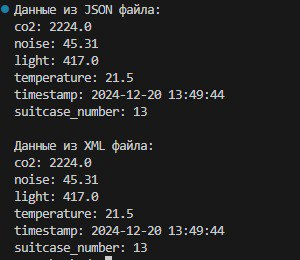


**Рисунок 7.5 – Данные через 10 секунд после запуска скрипта (xml файл)**

**Часть 3. Создание парсера**

*Листинг Б – Парсер скрипт*

|  |
| --- |
| import json  import xml.etree.ElementTree as ET  # Функция для чтения данных из JSON файла  def read\_from\_json(file\_path):  try:  with open(file\_path, 'r') as json\_file:  data = json.load(json\_file)  print("Данные из JSON файла:")  for key, value in data.items():  print(f"{key}: {value}")  except Exception as e:  print(f"Ошибка при чтении JSON файла: {e}")  # Функция для чтения данных из XML файла  def read\_from\_xml(file\_path):  try:  tree = ET.parse(file\_path)  root = tree.getroot()  print("\nДанные из XML файла:")  for child in root:  print(f"{child.tag}: {child.text}")  except Exception as e:  print(f"Ошибка при чтении XML файла: {e}")  # Пути к файлам JSON и XML  json\_file\_path = "data.json"  xml\_file\_path = "data.xml"  # Вызов функций для чтения данных  read\_from\_json(json\_file\_path)  read\_from\_xml(xml\_file\_path**)** |



**Рисунок 7.6 – Работа парсер скрипта**

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

**Цель работы:**

Цель данной работы – разработать программное обеспечение для автоматизированного сбора данных с датчиков с использованием MQTT-протокола и их дальнейшего анализа с помощью различных типов визуализации.

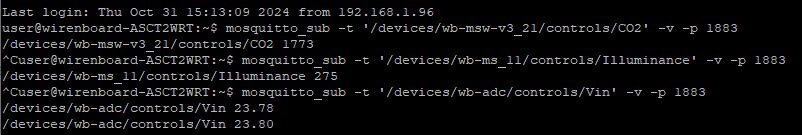
**Вариант № 5:**

1. Датчик устройства WB-MSW v.3 (5);

2. Датчик освещенности устройства WB-MS v.2 (12);

3. Напряжение на любом устройстве стенда.

**Часть 1. Подписка на топики**



**Рисунок 8.1 – Подписка на все топики по варианту**

**Часть 2. Основной скрипт**

*Листинг В – Основной скрипт для формирования двух файлов*

|  |
| --- |
| import csv  import time  import paho.mqtt.client as mqtt  # Настройки подключения к MQTT  broker\_address = "192.168.1.13"  port = 1883  topics = ["/devices/wb-msw-v3\_21/controls/CO2",  "/devices/wb-ms\_11/controls/Illuminance",  "/devices/wb-adc/controls/Vin"] |

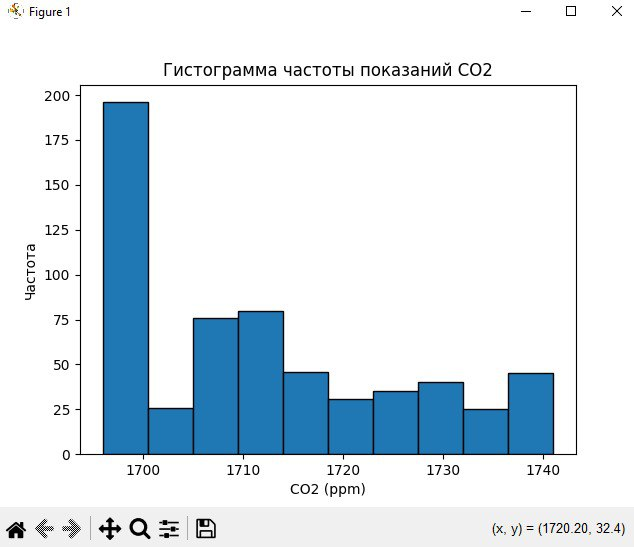
*Продолжение листинга В*

|  |
| --- |
| # Создание файла CSV  csv\_file = "sensor\_data.csv"  with open(csv\_file, mode='w', newline='') as file:  writer = csv.writer(file)  writer.writerow(["Timestamp", "CO2", "Illuminance", "Voltage"])  # Глобальные переменные для хранения данных  data = {"CO2": None, "Illuminance": None, "Voltage": None}  # Функция обработки сообщений  def on\_message(client, userdata, msg):  topic = msg.topic  value = float(msg.payload.decode())  if "/controls/CO2" in topic:  data["CO2"] = value  elif "/controls/Illuminance" in topic:  data["Illuminance"] = value  elif "/controls/Voltage" in topic:  data["Voltage"] = value  # Подключение к MQTT и подписка на топики  client = mqtt.Client()  client.on\_message = on\_message  client.connect(broker\_address, port, 60)  for topic in topics:  client.subscribe(topic)  client.loop\_start()  # Сбор данных в течение 10 минут  start\_time = time.time()  while time.time() - start\_time < 600:  if None not in data.values():  with open(csv\_file, mode='a', newline='') as file:  writer = csv.writer(file)  writer.writerow([time.strftime("%Y-%m-%d%H:%M:%S"), data["CO2"], data["Illuminance"], data["Voltage"]])  time.sleep(1) # Сохраняем данные каждую секунду  client.loop\_stop()  print("Сбор данных завершен. Данные сохранены в", csv\_file) |

**Часть 3. Скрипты для получения диаграмм**

*Листинг Г – Скрипт для создания гистограммы*

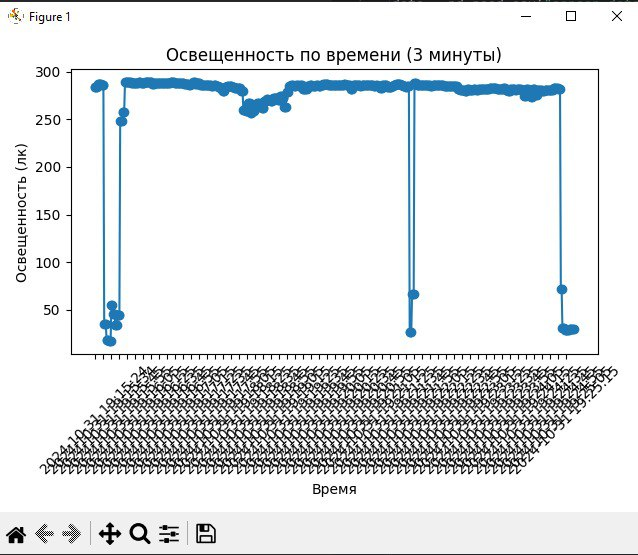
|  |
| --- |
| import pandas as pd  import matplotlib.pyplot as plt  data = pd.read\_csv("sensor\_data.csv")  plt.hist(data["CO2"], bins=10, edgecolor='black')  plt.title("Гистограмма частоты показаний CO2")  plt.xlabel("CO2 (ppm)")  plt.ylabel("Частота")  plt.show() |



**Рисунок 8.2 – Полученная гистограмма**

*Листинг Д – Скрипт для линейного графика*

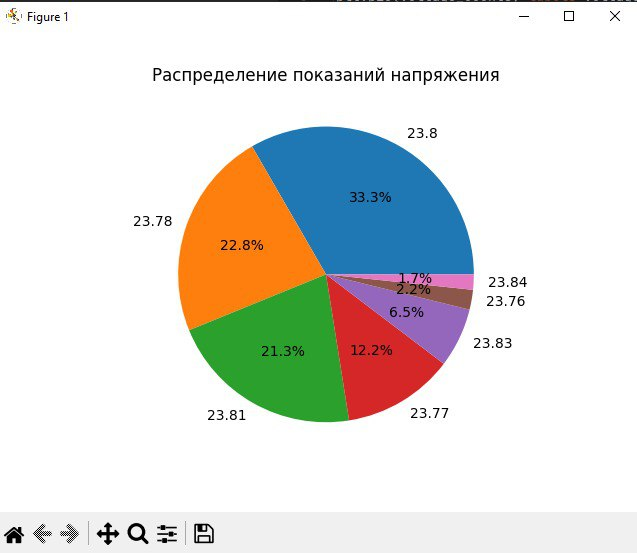
|  |
| --- |
| import pandas as pd  import matplotlib.pyplot as plt  data = pd.read\_csv("sensor\_data.csv")  plt.plot(data["Timestamp"], data["Illuminance"], marker='o')  plt.title("Освещенность по времени (3 минуты)")  plt.xlabel("Время")  plt.ylabel("Освещенность (лк)")  plt.xticks(data["Timestamp"][::10], rotation=45) # Отображение каждой 10-й метки  plt.tight\_layout()  plt.show() |



**Рисунок 8.3 – Полученный линейный график**

*Листинг Е – Скрипт для круговой диаграммы*

|  |
| --- |
| import pandas as pd  import matplotlib.pyplot as plt  data = pd.read\_csv("sensor\_data.csv")  voltage\_counts = data["Voltage"].value\_counts()  plt.pie(voltage\_counts, labels=voltage\_counts.index, autopct='%1.1f%%')  plt.title("Распределение показаний напряжения")  plt.show() |



**Рисунок 8.4 – Полученная круговая диаграмма**

**ДОПОЛНИТЕЛЬНЫЕ ЗАДАНИЯ**

**Дополнительное задание по практической работе № 5**

**Часть 1. Оборудование для сборки аппаратных компонентов системы климат-контроля умного дома**

Для успешной реализации системы климат-контроля умного дома необходимо выбрать надежные и точные датчики и актуаторы. Ниже представлены основные компоненты системы с подробным описанием каждого устройства:

**1. Датчики температуры**

1. DS18B20

* Название устройства: DS18B20 Температурный датчик
* Тип измерения: Цифровой
* Измеряемые параметры и диапазон: Температура от -55°C до +125°C
* Точность: ±0.5°C в диапазоне от -10°C до +85°C
* Напряжение питания: 3.0V - 5.5V
* Протокол передачи данных: 1-Wire
* Интерфейс управления: 1-Wire шина
* Описание входов и выходов, схема подключения:
  + - Входы: Питание (VCC), земля (GND), сигнал (DATA)
    - Выходы: Цифровой сигнал температуры
    - Схема подключения: Подключается к микроконтроллеру через 1-Wire шину с использованием резистора подтяжки 4.7 кОм между VCC и DATA.

2. DHT22

* Название устройства: DHT22 Температурно-влажностный датчик
* Тип измерения: Цифровой
* Измеряемые параметры и диапазон: Температура от -40°C до +80°C; Влажность от 0% до 100%
* Точность: ±0.5°C для температуры; ±2-5% для влажности
* Напряжение питания: 3.3V - 6V
* Протокол передачи данных: Односторонняя цифровая связь
* Интерфейс управления: Однопроводный (One-Wire) интерфейс
* Описание входов и выходов, схема подключения:
  + - Входы: Питание (VCC), земля (GND), сигнал (DATA)
    - Выходы: Цифровые данные температуры и влажности
    - Схема подключения: Подключается к микроконтроллеру через DATA пин с использованием резистора подтяжки 10 кОм между VCC и DATA.

**2. Датчики влажности**

1. HIH-4030

* Название устройства: HIH-4030 Датчик влажности и температуры
* Тип измерения: Аналоговый
* Измеряемые параметры и диапазон: Влажность от 0% до 100%; Температура от -40°C до +125°C
* Точность: ±3% для влажности; ±1°C для температуры
* Напряжение питания: 5V
* Протокол передачи данных: Аналоговый выход (вариабельное сопротивление)
* Интерфейс управления: Аналоговый интерфейс
* Описание входов и выходов, схема подключения:
  + - Входы: Питание (VCC), земля (GND)
    - Выходы: Аналоговый сигнал влажности и температуры
    - Схема подключения: Подключается к аналоговому входу микроконтроллера с использованием делителя напряжения для преобразования сигнала в цифровой формат через АЦП.

**3. Датчики качества воздуха**

1. MQ-135

* Название устройства: MQ-135 Датчик качества воздуха
* Тип измерения: Аналоговый и цифровой
* Измеряемые параметры и диапазон: Аммиак (NH₃), оксиды азота (NOx), алкоголей, бензолов, дыма; концентрации до 1000 ppm
* Точность: ±5% при калибровке
* Напряжение питания: 5V
* Протокол передачи данных: Аналоговый выход и цифровой выход через comparator
* Интерфейс управления: Аналоговый и цифровой интерфейс
* Описание входов и выходов, схема подключения:
  + - Входы: Питание (VCC), земля (GND)
    - Выходы: Аналоговый сигнал, цифровой сигнал через встроенный comparator
    - Схема подключения: Аналоговый выход подключается к АЦП микроконтроллера для считывания уровня загрязнителей, цифровой выход используется для сигнализации превышения пороговых значений.

**4. Актуаторы**

1. Реле модуль 5V

* Название устройства: Реле модуль SRD-05VDC-SL-C
* Тип измерения: Актуатор (электромеханическое)
* Измеряемые параметры и диапазон: Управление нагрузками до 10A при 250V AC или 30V DC
* Точность: Зависит от источника управления, механическое переключение
* Напряжение питания: 5V
* Протокол передачи данных: Электрический сигнал управления (LOW/HIGH)
* Интерфейс управления: GPIO пины микроконтроллера
* Описание входов и выходов, схема подключения:
  + - Входы: Управляющий сигнал от микроконтроллера, VCC и GND для питания
    - Выходы: Контактные клеммы для подключения управляемых нагрузок (NO, NC, COM)
    - Схема подключения: Управляющий сигнал подключается к IN пину реле через транзистор для усиления сигнала, VCC и GND обеспечивают питание реле. Нагрузки подключаются к контактам NO и COM.

2. Смарт-коннектор Wi-Fi (например, Sonoff Basic)

* Название устройства: Sonoff Basic Wi-Fi Smart Switch
* Тип измерения: Актуатор (смарт-выключатель)
* Измеряемые параметры и диапазон: Управление нагрузками до 16A при 250V AC
* Точность: Электронное переключение через Wi-Fi
* Напряжение питания: 100V - 240V AC
* Протокол передачи данных: Wi-Fi (IEEE 802.11 b/g/n)
* Интерфейс управления: Wi-Fi интерфейс, управление через мобильное приложение или облачную платформу
* Описание входов и выходов, схема подключения:
  + - Входы: Входное напряжение (L и N)
    - Выходы: Выходное напряжение (OUT и N)
    - Схема подключения: Входные клеммы подключаются к источнику питания, выходные клеммы – к управляемой нагрузке. Управление осуществляется через Wi-Fi, без необходимости физического подключения к микроконтроллеру.

**5. Контроллеры**

a. ESP32 Dev Kit

* Название устройства: ESP32 Dev Kit C
* Тип измерения: Микроконтроллер с встроенными Wi-Fi и Bluetooth модулями
* Измеряемые параметры и диапазон: Поддержка множества датчиков и актуаторов
* Точность: Высокая точность обработки данных, зависит от подключенных датчиков
* Напряжение питания: 5V (через USB) или 3.3V (логическая часть)
* Протокол передачи данных: Wi-Fi, Bluetooth, UART, SPI, I2C
* Интерфейс управления: GPIO, аналоговые и цифровые интерфейсы
* Описание входов и выходов, схема подключения:
  + - Входы: Подключение различных датчиков через GPIO, I2C, SPI интерфейсы
    - Выходы: Управление актуаторами через GPIO, PWM для регулировки устройств
    - Схема подключения: Датчики подключаются к соответствующим пинам ввода, актуаторы – к GPIO через драйверы или реле. Питание обеспечивает USB или внешний источник 5V, логическая часть работает на 3.3V.

**Целесообразность выбора оборудования**

1. DS18B20 и DHT22: Выбор этих датчиков обусловлен их высокой точностью, широким диапазоном измерений и цифровыми интерфейсами, которые облегчают интеграцию с микроконтроллером ESP32. DS18B20 идеален для точного измерения температуры, а DHT22 обеспечивает комбинированное измерение температуры и влажности.
2. HIH-4030: Этот аналоговый датчик влажности обладает хорошей точностью и широким диапазоном измерений, что позволяет создавать надежную систему контроля влажности в помещении.
3. MQ-135: Датчик качества воздуха MQ-135 способен обнаруживать различные загрязнители, обеспечивая безопасность и комфорт в помещении. Его аналоговый и цифровой выходы позволяют гибко настраивать систему оповещений и автоматического управления вентиляцией.
4. Реле модуль SRD-05VDC-SL-C и Sonoff Basic: Эти актуаторы позволяют управлять различными бытовыми устройствами (кондиционерами, обогревателями, вентиляторами) через микроконтроллер или напрямую через Wi-Fi, обеспечивая гибкость и масштабируемость системы.
5. ESP32 Dev Kit: Выбор ESP32 обусловлен его мощностью, наличием встроенных Wi-Fi и Bluetooth модулей, а также поддержкой множества интерфейсов, что делает его идеальным центральным контроллером для системы климат-контроля.

**Часть 2. Технологии передачи данных от физического устройства в Интернет**

**Выбор технологии: Wi-Fi (IEEE 802.11 b/g/n)**

**Обоснование выбора:**

1. **Скорость передачи данных:**
   * Wi-Fi обеспечивает высокую скорость передачи данных (до 600 Мбит/с для современных стандартов), что позволяет быстро передавать большие объемы информации, например, данные с нескольких датчиков и актуаторов.
2. **Объемы передаваемых данных:**
   * В системах климат-контроля объем передаваемых данных относительно невелик (температура, влажность, качество воздуха), что делает Wi-Fi избыточным по пропускной способности, но обеспечивающим высокую надежность и скорость передачи.
3. **Частота обновления:**
   * Wi-Fi поддерживает частые обновления данных, что важно для своевременного реагирования на изменения условий в помещении и обеспечения комфортного микроклимата.
4. **Энергоэффективность:**
   * Хотя Wi-Fi потребляет больше энергии по сравнению с другими технологиями (например, Zigbee или LoRa), современные модули, такие как ESP32, оптимизированы для энергосбережения. Для систем, подключенных к постоянному источнику питания (например, домашним сетям), это не является критичным.
5. **Доступность и инфраструктура:**
   * Wi-Fi является широко распространенной технологией с хорошо развитой инфраструктурой в большинстве домов и квартир, что упрощает интеграцию устройств в существующую сеть.
6. **Простота настройки и интеграции:**
   * Модули Wi-Fi, такие как ESP32, легко настраиваются и интегрируются с облачными платформами (например, ThingsBoard), обеспечивая бесперебойную связь между устройствами и сервером.
7. **Поддержка облачных платформ:**
   * Wi-Fi отлично сочетается с выбранной облачной платформой ThingsBoard, позволяя эффективно передавать данные в реальном времени и получать команды от облака.

**Сравнение с другими технологиями:**

1. Zigbee:
   * Преимущества: Низкое энергопотребление, хорошая поддержка сетей с большим количеством устройств.
   * Недостатки: Низкая скорость передачи данных, требует наличия шлюза для подключения к интернету.
2. Bluetooth:
   * Преимущества: Низкое энергопотребление, простота подключения.
   * Недостатки: Ограниченный радиус действия, менее подходит для сетей с множеством устройств.
3. LoRa:
   * Преимущества: Очень низкое энергопотребление, большой радиус действия.
   * Недостатки: Низкая скорость передачи данных, сложность настройки для домашних условий.
4. Ethernet:
   * Преимущества: Высокая надежность и скорость, низкая задержка.
   * Недостатки: Требует проводного подключения, ограниченная мобильность устройств.

**Дополнительное задание по практической работе № 6**

**Часть 1. Выбор облачной платформы для системы климат-контроля умного дома**

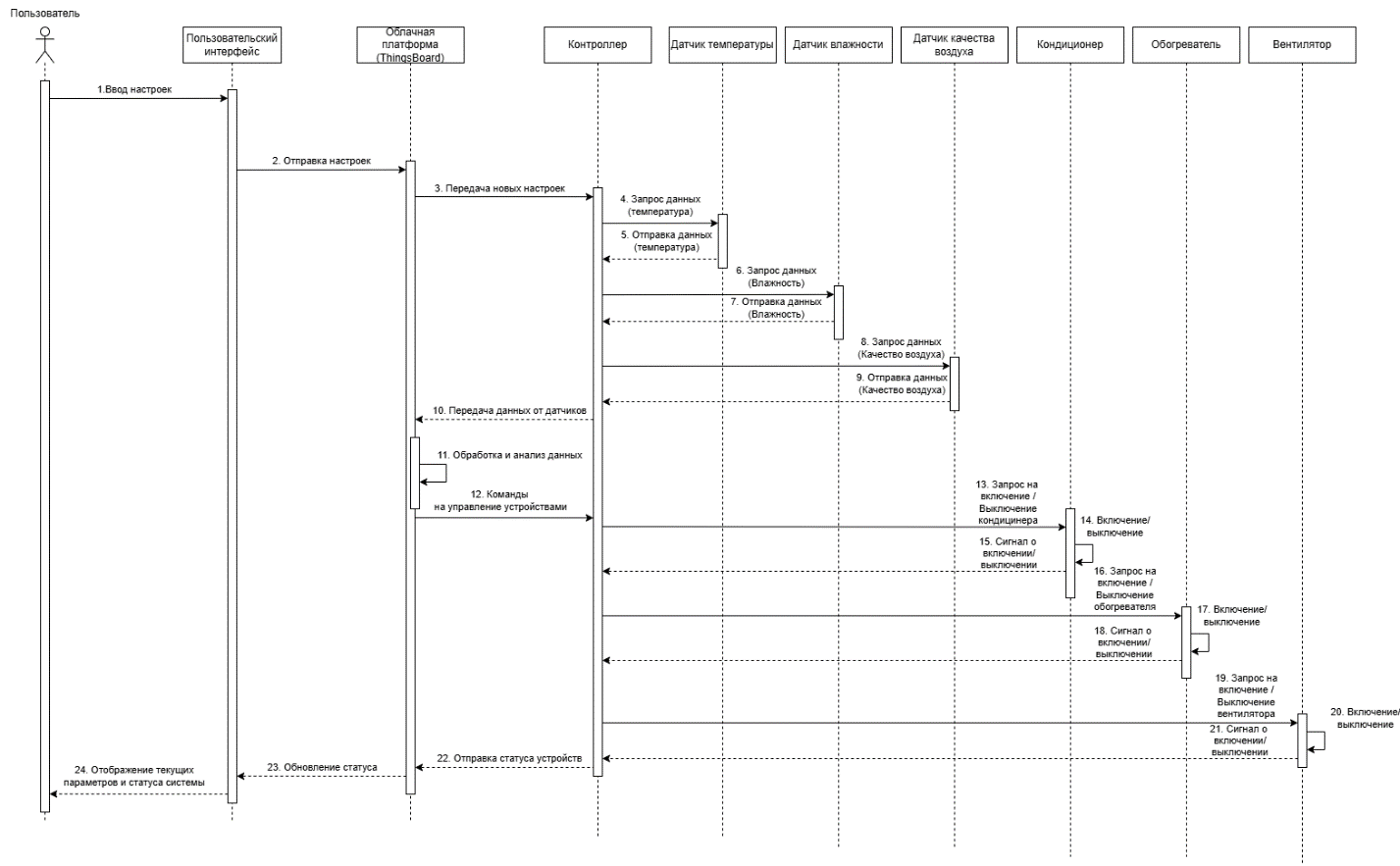
После тщательного анализа предложенных облачных платформ для реализации системы климат-контроля умного дома, наиболее подходящим выбором является [ThingsBoard](https://thingsboard.io/). Ниже представлены причины выбора именно этой платформы:

1. **Open-Source и Бесплатность:**
   * ThingsBoard является открытой платформой с возможностью бесплатного использования, что идеально подходит для проектов с ограниченным бюджетом. Это позволяет настраивать и расширять функционал без дополнительных затрат на лицензии.
2. **Гибкость и Масштабируемость:**
   * Платформа поддерживает управление большим количеством устройств, что обеспечивает возможность масштабирования проекта от небольших установок до крупных систем.
   * Позволяет создавать сложные правила автоматизации и сценарии взаимодействия между устройствами.
3. **Поддержка Разнообразных Датчиков и Устройств:**
   * ThingsBoard совместима с множеством протоколов передачи данных (MQTT, HTTP, CoAP), что облегчает интеграцию различных датчиков и актуаторов.
   * Легко добавлять новые устройства благодаря модульной архитектуре платформы.
4. **Мощные Инструменты для Визуализации и Анализа Данных:**
   * Предоставляет разнообразные виджеты и панели управления для визуализации данных в реальном времени.
   * Встроенные инструменты аналитики позволяют обрабатывать и анализировать собранные данные для принятия информированных решений.
5. **Активное Сообщество и Хорошая Документация:**
   * Широкое сообщество пользователей и разработчиков обеспечивает доступ к множеству примеров, руководств и решений распространённых проблем.
   * Подробная документация упрощает процесс развертывания и настройки платформы.
6. **Безопасность:**
   * Поддержка аутентификации и авторизации пользователей, шифрование данных, что обеспечивает высокий уровень безопасности системы.
7. **Интеграция с Экологическими Технологиями:**
   * Возможность интеграции с различными сервисами и технологиями, такими как солнечная энергия, что способствует повышению энерго-эффективности системы.

**Сравнение с Другими Платформами**

* “Microsoft Azure IoT” предлагает более широкий спектр сервисов и интеграций, но его стоимость может быть высокой для небольших проектов, а также требует значительных навыков для эффективного использования.
* “OpenHAB” тоже является отличной open-source платформой для домашней автоматизации, но ThingsBoard предоставляет более обширные возможности для анализа данных и визуализации, что важно для системы климат-контроля.
* “AdaFruit”, “HiveMQ”, “Yandex Cloud” и другие платформы имеют свои преимущества, но ThingsBoard сочетает в себе баланс между функциональностью, гибкостью и стоимостью, делая её наиболее подходящей для данного проекта.

**Часть 2. Построение диаграммы последовательности**

****

**Рисунок 6.6 – Взаимодействие всех компонентов проекта**

**ВЫВОД**

Выполненные лабораторные работы продемонстрировали глубокое изучение измерительных и исполнительных устройств в системах Интернета вещей. Были рассмотрены различные типы датчиков и устройств. Каждый из них был проанализирован в контексте характеристик, принципов работы и способов интеграции в систему автоматизации. Исследованы цифровые и аналоговые параметры, описаны диапазоны измерения, а также рассмотрены схемы подключения и интерфейсы управления.

Был также разработан ряд программных решений для автоматизированного сбора, анализа и визуализации данных. Использование протокола MQTT позволило организовать подписку на топики для получения данных с различных датчиков в реальном времени.

Для последующего анализа собранных данных были созданы сценарии для их визуализации с помощью различных типов диаграмм: гистограммы, линейного графика и круговой диаграммы. Визуализация данных помогла наглядно продемонстрировать частотное распределение показаний , динамику освещенности во времени и распределение значений напряжения.

Таким образом, работы способствовали не только освоению принципов подключения и управления устройствами, но и применению методов анализа данных в контексте IoT-систем.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Modbus: Modbus Organization. "Modbus Protocol Specification." [www.modbus.org](https://www.modbus.org)
2. 1-Wire: Maxim Integrated. "1-Wire Technology Overview." [www.maximintegrated.com](http://www.maximintegrated.com)
3. Официальный документ от NXP, описывающий спецификации – https://www.i2c-bus.org/specification/
4. CAN Products by Microchip — страница с информацией о продукции с поддержкой CAN – https://www.microchip.com/en-us/products/interface-and-connectivity/can
5. ThingsBoard Documentation. "Getting Started with ThingsBoard." <https://thingsboard.io/docs/>
6. ThingsBoard Academy. "IoT and ThingsBoard Training Resources." https://thingsboard.io/academy/