Lab3

ссылка на github ; https://github.com/Muhammeddada120/Labs\_internet-Things

Аттиа Мохамед Магди Абдельхамид Мохамед

Отчёт

# Цель

Цель данной лабораторной работы — разработать Телеграм-бота, который может взаимодействовать с сервером MQTT. Бот позволяет пользователям:

* Получать телеметрические данные от датчиков.
* Управлять исполнительным устройством в ручном режиме через команды Телеграм.

# Описание разработанной программы

Программа объединяет функциональность Телеграм и MQTT, обеспечивая взаимодействие с IoT-устройствами в реальном времени. Основные возможности включают:

* **Передача телеметрических данных**: Бот передаёт данные датчиков, полученные с MQTT-брокера.
* **Управление исполнительным устройством**: Пользователи могут отправлять команды для управления состоянием исполнительного устройства (например, включение/выключение насоса).

# Функционал

1. **Передача телеметрии**:
   * Данные датчиков публикуются в MQTT-брокер и отображаются пользователю по запросу.
2. **Управление исполнительным устройством**:
   * Пользователи могут вручную активировать или деактивировать исполнительное устройство через команды Телеграм.
3. **Используемые топики**:
   * iot/sensor/soil\_moisture: Топик для данных с датчиков.
   * iot/actuator/pump\_command: Топик для управления исполнительным устройством.

# Возможности Телеграм-бот

* Создан с использованием библиотеки python-telegram-bot.
* Поддерживает следующие команды:
  + /start: Отображает приветственное сообщение и список доступных команд.
  + /status: Показывает текущие данные датчиков и состояние исполнительного устройства.
  + /start\_pump: Включает исполнительное устройство (насос).
  + /stop\_pump: Выключает исполнительное устройство (насос).

# Интеграция с MQTT

* **Брокер**: Используется публичный MQTT-сервер test.mosquitto.org.
* **Клиент**: Подключается к брокеру для публикации и подписки на топики.
* **Многопоточность**: Обеспечивает параллельную работу MQTTцикла и Телеграм-бота.

# Реализация •Язык программирования: Python

•**Библиотеки**:

* paho-mqtt для работы с MQTT.
* python-telegram-bot для интеграции с Телеграм.
* threading для многопоточности.

# Обзор кода

1. **Настройка MQTT**:
   * Подключение к брокеру. oОбработка подписки и сообщений для телеметрических данных.
2. **Команды Телеграм**:
   * Обработка пользовательских команд для получения данных датчиков или управления исполнительным устройством.
3. **Интеграция**:
   * Объединение функционала MQTT и Телеграм для обеспечения удобного взаимодействия.

# Результаты Успешные результаты

* Данные датчиков успешно передаются от MQTT-брокера в Телеграм-бот.
* Команды для управления исполнительным устройством (включение/выключение) обрабатываются через MQTT-брокер.
* Бот отвечает на пользовательские команды в режиме реального времени.

# Наблюдения

* Телеметрические данные динамически обновляются в боте.
* Состояние исполнительного устройства меняется сразу после отправки команды пользователем.

# Проблемы и их решения Проблема 1: Изменения в API Телеграм-бота

* **Проблема**: Устаревший класс Updater вызывал ошибки при инициализации.
* **Решение**: Обновлён код для использования нового класса Application из библиотеки python-telegram-bot (версии 20+).

# Проблема 2: Параллельная работа MQTT и Телеграм

* **Проблема**: Блокирующее поведение метода MQTT loop\_forever().
* **Решение**: Реализована многопоточность для независимой работы MQTT-цикла и Телеграм-бота.

# Проблема 3: Отображение данных в реальном времени

* **Проблема**: Обеспечение своевременного обновления данных датчиков.
* **Решение**: Оптимизирована обработка сообщений MQTT и ответы Телеграм-бота для достижения высокой производительности.

# Как запустить программу Требования

1. Установить Python 3.9 или более позднюю версию.
2. Установить необходимые библиотеки:

**pip install paho-mqtt python-telegram-bot==20.0**

1. Получить токен Телеграм-бота от BotFather.

# Шаги для выполнения

1. Сохраните программу как telegram\_mqtt\_bot.py.
2. Замените YOUR\_TELEGRAM\_BOT\_TOKEN на реальный токен бота в коде.
3. Запустите программу:

**python telegram\_mqtt\_bot.py**

1. Взаимодействуйте с ботом через Телеграм, используя следующие команды:

* /start: Просмотр инструкций.
* /status: Получение текущих данных датчиков и состояния исполнительного устройства. o/start\_pump: Включение насоса.
* /stop\_pump: Выключение насоса.

# Заключение

Разработанный Телеграм-бот успешно интегрирован с сервером MQTT для передачи телеметрии и управления исполнительным устройством. Проект демонстрирует практическое применение технологий IoT и обмена сообщениями для эффективного удалённого мониторинга и управления.