|  |  |  |
| --- | --- | --- |
|  |  |  |
| 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-36-22 | Казаков И. Г.  Утенков Ю. Ю.  Ярош В. Э.  Захаров А. А.  Аскаров Т. Е. |
| **Проверил:** | Образцов В. М. |

2024 г.

Оглавление

[1. Цель работы 3](#_Toc180697988)

[1.1. Задание для выполнения 3](#_Toc180697989)

[1.2. Команда 3](#_Toc180697990)

[2. Ход выполнения работы 3](#_Toc180697991)

[2.1. Практическая работа №5 3](#_Toc180697992)

[2.4. Практическая работа №6 20](#_Toc180697993)

[2.5. Практическая работа №7 24](#_Toc180697994)

[2.6. Практическая работа №8 27](#_Toc180697995)

[3. Вывод 32](#_Toc180697996)

[4. Список использованных источников 32](#_Toc180697997)

# **Цель работы**

## Задание для выполнения

Блок практических работ состоит из 4 заданий, каждое задание предполагает взаимодействие со стендом и оборудованием в его составе. Работы включают в себя как физическое взаимодействие с приборами, так и написание программного кода. В отчете по практическим заданиям представлены процесс и результат выполнения в виде выводов и скриншотов работы стенда.

## Команда

Состав команды: Казаков И. Г., Утенков Ю. Ю., Ярош В. Э., Захаров А. А., Аскаров Т. Е.

# **Ход выполнения работы**

## Практическая работа №5

* + 1. Постановка задачи

Ознакомиться с устройствами в составе стенда и их характеристиками.

Описать датчики и исполнительные устройства, согласно варианту:

1. Датчик температуры 1-wire DS18B20 (2)
2. Уровень шума в составе устройства WB-MSW v.3 (5)
3. Модуль реле 3-канальный WB-MR3 (21)

Описать принцип работы, преимущества и недостатки, сферу применения следующих четырех технологий:

1. Modbus RTU;
2. 1-Wire;
3. I²C;
4. CAN.
   * 1. Выполнение задачи
   1. **Часть 1. Измерительные и исполнительные устройства стенда**

**Датчик температуры 1-wire DS18B20 (2)**

Датчик – чувствительный элемент, который контактирует с окружающей средой: измеряет показания, реагирует на объекты и создает сигнал об этом.

Датчик температуры 1-wire DS18B20 - это цифровой измеритель температуры, с разрешением преобразования 9 - 12 разрядов и функцией тревожного сигнала контроля за температурой. Параметры контроля могут быть заданы пользователем и сохранены в энергонезависимой памяти датчика.

DS18B20 обменивается данными с микроконтроллером по однопроводной линии связи, используя протокол интерфейса 1-Wire.

Питание датчик может получать непосредственно от линии данных, без использования внешнего источника. В этом режиме питание датчика происходит от энергии, запасенной на паразитной емкости.

Диапазон измерения температуры составляет от -55 до +125 °C. Для диапазона от -10 до +85 °C погрешность не превышает 0,5 °C.

У каждой микросхемы DS18B20 есть уникальный серийный код длиной 64 разряда, который позволяет нескольким датчикам подключаться на одну общую линию связи.

Через один порт микроконтроллера можно обмениваться данными с несколькими датчиками, распределенными на значительном расстоянии. Режим крайне удобен для использования в системах экологического контроля, мониторинга температуры в зданиях, узлах оборудования.

Вывод сигнала данных (входа/выход). Выход типа открытый коллектор интерфейса 1-Wire. Также через него происходит питание в режиме паразитное питание. Датчик имеет три вывода (Рисунок 1).

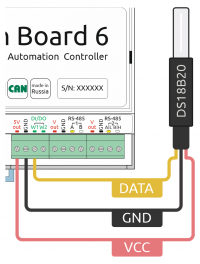


Рисунок 1 – Подключение датчика 1-Wire к контроллеру

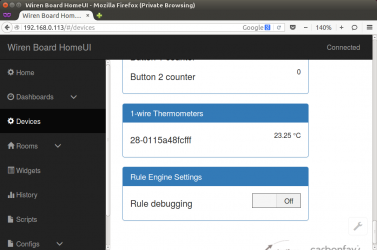


Рисунок 2 – Показания датчика и его уникальный идентификатор на странице Devices веб-интерфейса

**Уровень шума в составе устройства WB-MSW v.3 (5)**

Комбинированный цифровой датчик температуры, влажности, освещенности, движения, уровня шума, концентрации CO2 и летучих органических соединений. Оснащён ИК приемопередатчиком. Предназначен для контроля климата в жилых и офисных помещениях, для бытового использования. Датчик выполнен в пластиковом корпусе и предназначен для крепления на стену. Управление модулем производится с контроллера или ПК по шине RS-485 командами по протоколу Modbus RTU.

**Измеряемые величины:**

* Температура: -40°С — +80°С (+-0.5°С)
* Влажность: 0 — 98% (+-3%)
* Освещенность: 0.02 — 100000 лк
* Уровень шума: 38 — 105 дБА
* Концентрация СО2: 400 — 5000 ppm
* Концентрация VOC: 0 — 60000 ppb
* Движение: до 8 м, угол - 120°

**Дополнительные функции:**

* Передача ИК-команд
* Управляемый зуммер
* Двухцветная индикация, управляемая по Modbus
* Подогрев датчика для работы в условиях высокой влажности

**Прочее:**

* Напряжение питания: от 9 до 28 В постоянного тока
* Интерфейс: RS-485, Modbus RTU
* Пластиковый корпус с возможностью крепления на стену (83x83x21 мм)

**Датчик шума**

В модуле используется микрофон с усилителем и фильтрами для коррекции по шкале А с учётом особенностей восприятия человеческим ухом звуков разных частот. Шум измеряется в акустических децибелах (дБА), что позволяет контролировать шумовую обстановку в соответствии со стандартами и санитарными нормами.

**Управление устройством и просмотр значений**

В веб-интерфейсе вы можете управлять выходами устройства и просматривать полученные с него значения. Список отображаемых каналов можно изменить через настройки устройства, доступные на [странице выбора шаблона](https://wirenboard.com/wiki/RS-485:Configuration_via_Web_Interface#%D0%9E%D1%82%D0%BA%D0%BB%D1%8E%D1%87%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BA%D0%B0%D0%BD%D0%B0%D0%BB%D0%BE%D0%B2,_%D1%81%D0%BC%D0%B5%D0%BD%D0%B0_%D1%80%D0%B5%D0%B6%D0%B8%D0%BC%D0%BE%D0%B2_%D0%B8_%D0%BD%D0%B0%D1%81%D1%82%D1%80%D0%BE%D0%B9%D0%BA%D0%B0_%D0%BF%D0%B0%D) (Рисунок 3).

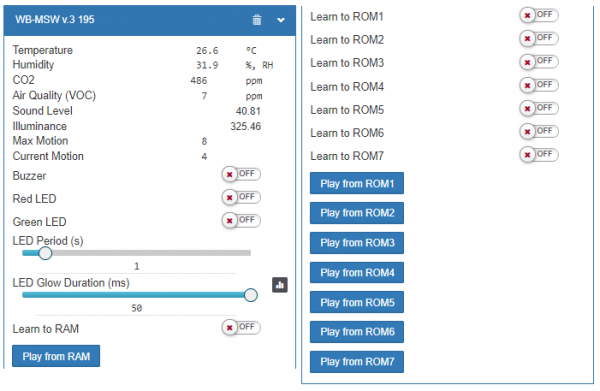
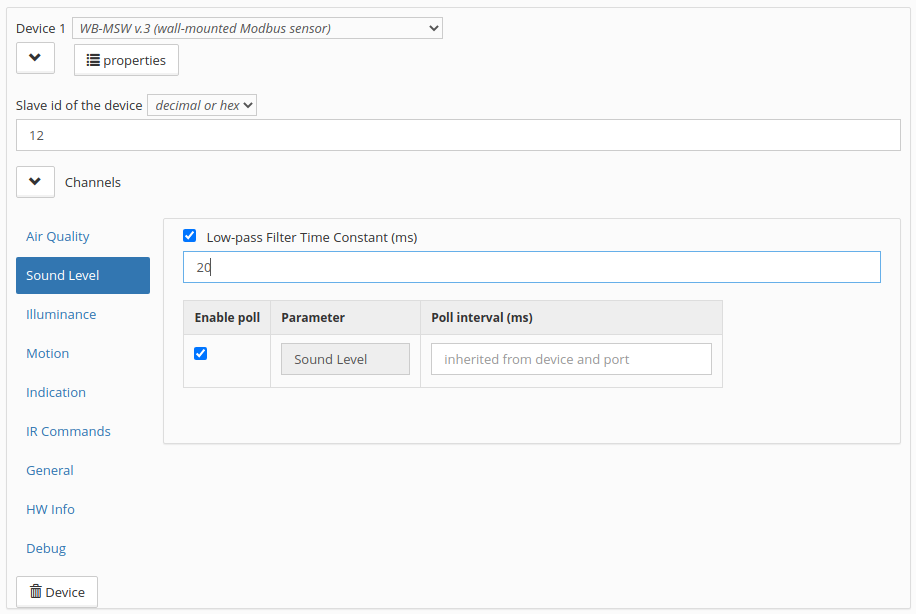


Рисунок 3 - Элементы управления и индикации модуля WB-MSW v.3 в веб-интерфейсе

**Настройка**

Способы настройки:

1. Указать параметры в веб-интерфейсе контроллера Wiren Board. Перейдите на страницу настройки serial-устройств, выберите порт, найдите или добавьте устройство и измените параметры. Если нужный параметр отсутствует в шаблоне, его можно задать через пользовательские параметры.
2. Записать настройки в Modbus-регистры модуля из консоли контроллера с помощью утилиты modbus­\_client.

****Рисунок 4 – Пример настройки датчика шума

* 1. **Часть 2. Протоколы работы с устройствами**

**Modbus RTU**

**Modbus** — коммуникационный протокол, основан на архитектуре ведущий-ведомый. Использует для передачи данных интерфейсы RS-485, RS-422, RS-232, а также Ethernet сети TCP/IP.

Если отбросить SlaveID адрес и CRC контрольную сумму, то получится PDU, Protocol Data Unit. SlaveID – это адрес устройства, может принимать значение от 0 до 247, адреса с 248 до 255 зарезервированы. Данные в модуле хранятся в 4 таблицах. Две таблицы доступны только для чтения и две для чтения-записи. В каждой таблице помещается 9999 значений. В сообщении Modbus используется адрес регистра. Например, первый регистр AO Holding Register, имеет номер 40001, но его адрес равен 0000. Разница между этими двумя величинами есть смещение offset.

Каждая таблица имеет свое смещение, соответственно: 1, 10001, 30001 и 40001. Регистр аналогового выхода AO0 имеет значение AE 41 HEX или 44609 в десятичной системе. Регистр аналогового выхода AO1 имеет значение 56 52 HEX или 22098 в десятичной системе. Регистр аналогового выхода AO2 имеет значение 43 40 HEX или 17216 в десятичной системе. Значение AE 41 HEX - это 16 бит 1010 1110 0100 0001, может принимать различное значение, в зависимости от типа представления. Значение регистра 40108 при комбинации с регистром 40109 дает 32 бит значение.

Программы для работы с протоколом Modbus RTU:

* DCON Utility Pro с поддержкой Modbus RTU, ASCII, DCON.
* Modbus Master Tool с поддержкой Modbus RTU, ASCII, TCP
* Modbus TCP client с поддержкой Modbus TCP.

**1-Wire**

1-Wire — двунаправленная шина связи для устройств с низкоскоростной передачей данных (обычно 15,4 Кбит/с, максимум 125 Кбит/с в режиме overdrive), в которой данные передаются по цепи питания (то есть всего используются два провода — один общий (GND), а второй для питания и данных; в некоторых случаях используют и отдельный провод питания). Разработана корпорацией Dallas Semiconductor (с 2001 года — Maxim Integrated) и является её зарегистрированной торговой маркой. Соответственно, топология такой сети — общая шина. Сеть устройств 1-Wire со связанным основным устройством названа «MicroLan», это также торговая марка Dallas Semiconductor. Обычно используется для того, чтобы связываться с недорогими простыми устройствами, такими, как, например, цифровые термометры и измерители параметров внешней среды.

Устройство 1-Wire может находиться как на печатной плате вместе с устройством управления, так и отдельно. Иногда они предназначены лишь для поддержки устройств 1-Wire, но во многих коммерческих приложениях устройство 1-Wire — просто один из чипов, создающих нужное решение. Иногда они присутствуют, например, в аккумуляторных батареях ноутбуков и сотовых телефонов. Некоторые лабораторные системы и другие системы сбора данных и управляющие системы подключают к устройствам 1-Wire, используя шнуры с модульными разъёмами или с кабелем CAT-5, с устройствами, установленными в разъём, включёнными в небольшую печатную плату, или присоединёнными к исследуемому объекту. В таких системах популярен разъём RJ11 (6P2C или модульные разъёмы 6P4C, обычно используемые для телефонов). Системы датчиков и приводов могут быть связаны компонентами 1-Wire, каждый из которых включает в себя всё необходимое для функционирования шины 1-Wire. В качестве примера можно привести термометрию, таймеры, датчики напряжений и токов, контролирование батарей, и память. Они могут быть подключены к ПК при помощи преобразователей шины. Последовательные интерфейсы USB, RS-232, и параллельный интерфейс (LPT) являются популярными решениями для соединения MicroLan с ПК. MicroLan также является интерфейсом для микроконтроллеров, таких, как Atmel AVR, Parallax BASIC Stamp и семейство Microchip PIC. Однако аппаратной поддержки этой шины микроконтроллеры (AVR, PIC и другие), как правило, не имеют, и работа с шиной реализуется программно, с использованием сторонних библиотек (вроде Arduino и других), либо программист, имея спецификацию, может разработать сам необходимую функциональность.

**Достоинства:**

* Для связи с устройством требуется лишь два провода: на данные и заземление. Интегральная схема включает конденсатор ёмкостью 800 пФ для питания от линии данных (так называемое паразитное питание).
* Большое расстояние передачи. Расстояние достигает 300 м при соблюдении ряда условий:
* применение кабеля типа «витая пара»;
* использование специального драйвера сети (активная подтяжка с учётом тока в линии);
* использование топологии «общая шина» с единым стволом (не свободная топология);
* Изменяемость конфигурации любой сети 1-Wire в процессе её работы.

**Идентификация личности**

Каждая микросхема 1-Wire имеет уникальный номер. Это позволяет использовать устройства iButton в качестве простых идентификаторов личности, например, в системах контроля и управления доступом (СКУД). В этом качестве они успешно конкурируют с бесконтактными карточками, использующими технологию RFID. Имеются устройства iButton с поддержкой криптографии, что позволяет создавать на их основе защищённые хранилища небольших объёмов данных или средства сильной аутентификации. Такие устройства могут конкурировать со смарт-картами в некоторых применениях.

**Удалённые датчики физических величин**

Устройства 1-Wire очень удобны для измерений. Не требуется отдельного питания, возможно подключить по одному проводу целую гирлянду разнообразных датчиков. Система таких датчиков легко контролируется на предмет аварий. Записи о калибровках могут храниться прямо в датчиках. Измерение температуры — одно из самых массовых применений 1-Wire устройств. В сельском хозяйстве применяется для многоточечного контроля температуры в теплицах, ульях, элеваторах, инкубаторах, овощехранилищах. Популярны домашние метеостанции, подключаемые по этому интерфейсу.

**Маркировка оборудования**

Микросхемы 1-Wire популярны для маркировки и хранения параметров дополнительного оборудования к установкам. Например, медицинские и лабораторные приборы, использующие в работе множество различных сменных головок и датчиков, снабжаются микросхемой. При подключении прибор сразу распознаёт сменную головку и корректно устанавливает режим работы. Аналогично может контролироваться наработка узлов с ограниченным ресурсом.

**I²C**

I²C **—** последовательная асимметричная шина для связи между интегральными схемами внутри электронных приборов. Использует две двунаправленные линии связи (SDA и SCL), применяется для соединения низкоскоростных периферийных компонентов с процессорами и микроконтроллерами (например, на материнских платах, во встраиваемых системах, в мобильных телефонах).Шина I2C синхронная, состоит из двух линий: данных (SDA) и тактирования (SCL). Есть ведущий (master) и ведомые (slave). Инициатором обмена всегда выступает ведущий, обмен между двумя ведомыми невозможен. Всего на одной двухпроводной шине может быть до 127 устройств. Такты на линии SCL генерирует master. Линией SDA могут управлять как мастер, так и ведомый в зависимости от направления передачи. Единицей обмена информации является пакет, обрамленный уникальными условиями на шине, именуемыми стартовым и стоповым условиями. Мастер в начале каждого пакета передает один байт, где указывает адрес ведомого и направление передачи последующих данных. Данные передаются 8-битными словами. После каждого слова передается один бит подтверждения приема приемной стороной.

I²C находит применение в устройствах, предусматривающих простоту разработки и низкую себестоимость изготовления при относительно неплохой скорости работы.

Список возможных применений:

* доступ к модулям памяти NVRAM;
* доступ к низкоскоростным ЦАП/АЦП;
* регулировка контрастности, насыщенности и цветового баланса мониторов;
* регулировка звука в динамиках;
* управление светодиодами, в том числе в мобильных телефонах;
* чтение информации с датчиков мониторинга и диагностики оборудования, например, термостат центрального процессора или скорость вращения вентилятора охлаждения;
* чтение информации с часов реального времени (кварцевых генераторов);
* управление включением/выключением питания системных компонент;
* информационный обмен между микроконтроллерами;

**Преимущества:**

* необходим всего один микроконтроллер для управления набором устройств;
* используется всего два проводника для подключения многих устройств;
* возможна одновременная работа нескольких ведущих (master) устройств, подключенных к одной шине I²C;
* стандарт предусматривает «горячее» подключение и отключение устройств в процессе работы системы;
* встроенный в микросхемы фильтр подавляет всплески, обеспечивая целостность данных.

**Недостатки:**

* ограничение на ёмкость линии — 400 пФ;
* несмотря на простоту протокола, программирование контроллера I²C затруднено из-за изобилия возможных нештатных ситуаций на шине. По этой причине большинство систем используют I²C c единственным ведущим (Master) устройством, и распространённые драйверы поддерживают только монопольный режим обмена по I²C;
* трудность локализации неисправности, если одно из подключенных устройств ошибочно устанавливает на шине состояние низкого уровня.

**CAN**

CAN — стандарт промышленной сети, ориентированный, прежде всего, на объединение в единую сеть различных исполнительных устройств и датчиков. Режим передачи — последовательный, широковещательный, пакетный.

CAN разработан компанией Robert Bosch GmbH в середине 1980-х и в настоящее время широко распространён в промышленной автоматизации, технологиях домашней автоматизации («умного дома»), автомобильной промышленности и многих других областях. Стандарт для автомобильной автоматики.

CAN является синхронной шиной с типом доступа Collision Resolving (CR, разрешение коллизии), который, в отличие от Collision Detect (CD, обнаружение коллизии) сетей (Ethernet), детерминировано (приоритетно) обеспечивает доступ на передачу сообщения, что особо ценно для промышленных сетей управления (fieldbus). Передача ведётся кадрами. Полезная информация в кадре состоит из идентификатора длиной 11 бит (стандартный формат) или 29 бит (расширенный формат, надмножество предыдущего) и поля данных длиной от 0 до 8 байт. Идентификатор говорит о содержимом пакета и служит для определения приоритета при попытке одновременной передачи несколькими сетевыми узлами.

Для абстрагирования от среды передачи спецификация CAN избегает описывать биты данных как «0» и «1». Вместо этого применяются термины «рецессивный» бит и «доминантный» бит, при этом подразумевается, что при передаче одним узлом сети рецессивного бита, а другим доминантного принят будет доминантный бит. Например, при реализации физического уровня на радиоканале отсутствие сигнала означает рецессивный бит, а наличие — доминантный; тогда как в типичной реализации проводной сети рецессив бывает при наличии сигнала, а доминант, соответственно, при отсутствии. Стандарт сети требует от «физического уровня», фактически, единственного условия: чтобы доминантный бит мог подавить рецессивный, но не наоборот. Например, в оптическом волокне доминантному биту должен соответствовать «свет», а рецессивному — «темнота».

В электрическом проводе может быть так: рецессивное состояние — высокое напряжение на линии (от источника с большим внутренним сопротивлением), доминантное — низкое напряжение (доминантный узел сети «подтягивает» линию на землю). Если линия находится в рецессивном состоянии, перевести её в доминантный может любой узел сети (включив свет в оптоволокне или закоротив высокое напряжение). Наоборот — нельзя (включить темноту нельзя).

**Преимущества:**

* Возможность работы в режиме жёсткого реального времени.
* Простота реализации и минимальные затраты на использование.
* Высокая устойчивость к помехам.
* Арбитраж доступа к сети без потерь пропускной способности.
* Надёжный контроль ошибок передачи и приёма.
* Широкий диапазон скоростей работы.
* Большое распространение технологии, наличие широкого ассортимента продуктов от различных поставщиков.

**Недостатки:**

* Небольшое количество данных, которое можно передать в одном пакете (до 8 байт).
* Большой размер служебных данных в пакете (по отношению к полезным данным).
* Отсутствие единого общепринятого стандарта на протокол высокого уровня.

## Практическая работа №6

* + 1. Постановка задачи

Подключитесь к консоли WirenBoard по протоколу SSH. Для этого используйте SSH-клиент PuTTy (или другой клиент). При помощи улиты mosquitto\_sub подпишитесь на сообщения нескольких датчиков стенда: температуры и движения.

Включите или измените поведение устройств посредством отправки сообщение в соответствующий топик согласно вариантам:

* + 1. Выполнение задачи
    2. Часть 1

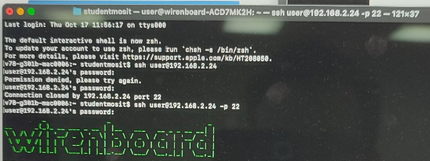


Рисунок 5 – Подключение Wirenboard по протоколу SSH

* + 1. Часть 2

При помощи улиты mosquitto\_sub подпишитесь на сообщения нескольких датчиков стенда, согласно индивидуальному варианту (рисунки 10 - 12).

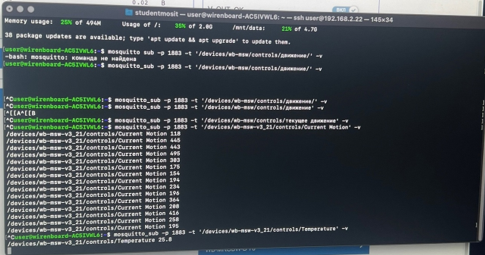


Рисунок 6 – подписка на сообщения датчиков стенда при помощи mosquitto\_sub, часть 1

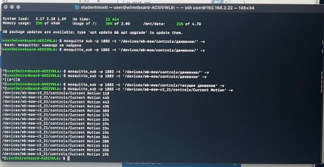


Рисунок 7 – подписка на сообщения датчиков стенда при помощи mosquitto\_sub, часть 2

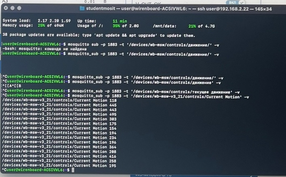


Рисунок 8 – подписка на сообщения датчиков стенда при помощи mosquitto\_sub, часть 3

* + 1. Часть 3

Включите или измените поведение устройств посредством отправки сообщение в соответствующий топик согласно вариантам (рисунки 9 - 12).

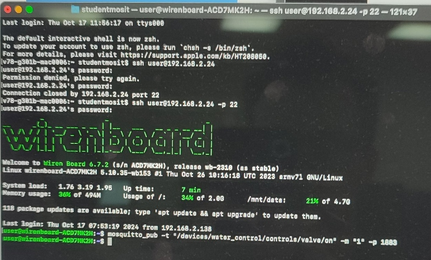


Рисунок 9 – включение подписки на получение данных с датчика контроля влажности

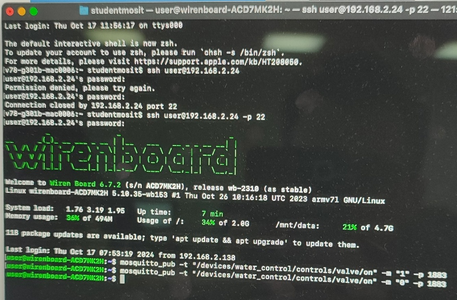


Рисунок 10 – выключение подписки на получение данных с датчика контроля влажности

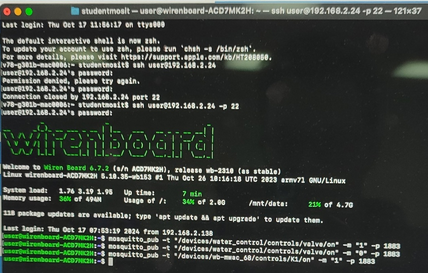


Рисунок 11 – включение подписки на получение данных с датчика wb-mac\_68

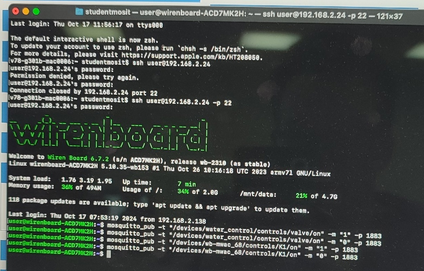


Рисунок 12 – выключение подписки на получение данных с датчика wb-mac\_68

## Практическая работа №7

* + 1. Постановка задачи

С компьютера в аудитории или личного устройства подпишитесь на несколько MQTT-топиков стенда согласно вариантам:

На любом языке программирования реализуйте программу (скрипт), которая бы каждые 5 секунд упаковывала последние полученные данные в файлы формата JSON и XML. В одной записи должно быть 6 полей: 4 показаний датчиков, время формирования файла, номер чемодана (последние две цифры IP-адреса).

3. На любом языке программирования реализуйте программу-парсер, которая бы выводила в консоль данные, полученные из сгенерированных в п.2 файлов.

|  |  |
| --- | --- |
| **№ Варианта** | **Датчики** |
| 2 | 1. Датчик шума устройства WB-MSW v.3 (5)  2. Датчик освещенности устройства WB-MS v.2 (12)  3. Датчик CO2 устройства WB-MSW v.3 (5)  4. Датчик температуры 1-wire DS18B20 (2) |

* + 1. Выполнение задачи
    2. Часть 1

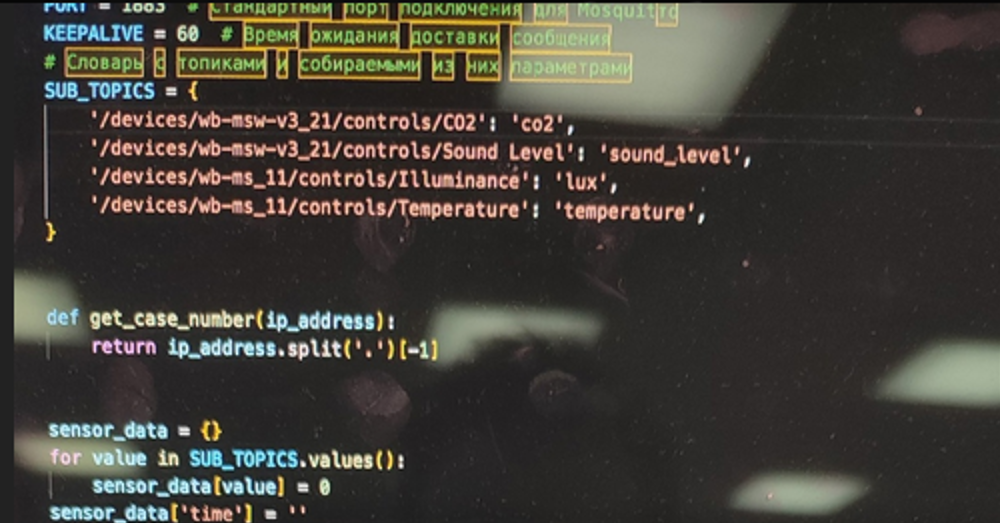


Рисунок 13 – Часть кода для подписки на рассылку сообщений и запись данных в файл

* + 1. Часть 2

На любом языке программирования реализуйте программу (скрипт), которая бы каждые 5 секунд упаковывала последние полученные данные в файлы формата JSON и XML. В одной записи должно быть 6 полей: 4 показаний датчиков, время формирования файла, номер чемодана (последние две цифры IP-адреса) (рисунки 14- 15).

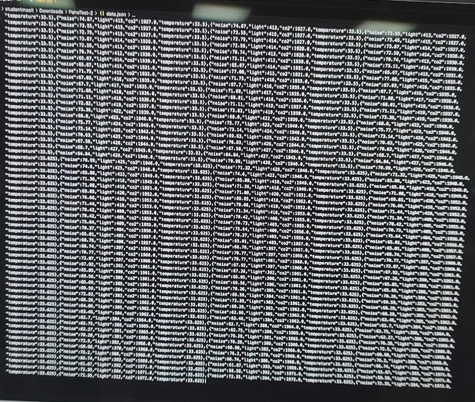


Рисунок 14 – Упаковка данных программой в файл формата JSON

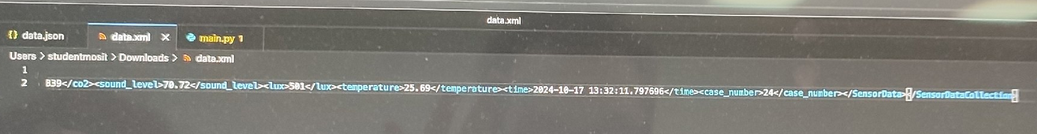


Рисунок 15 – Упаковка данных программой в файл формата XML

* + 1. Часть 3

На любом языке программирования реализуйте программу-парсер, которая бы выводила в консоль данные, полученные из сгенерированных в п.2 файлов (рисунок 16).

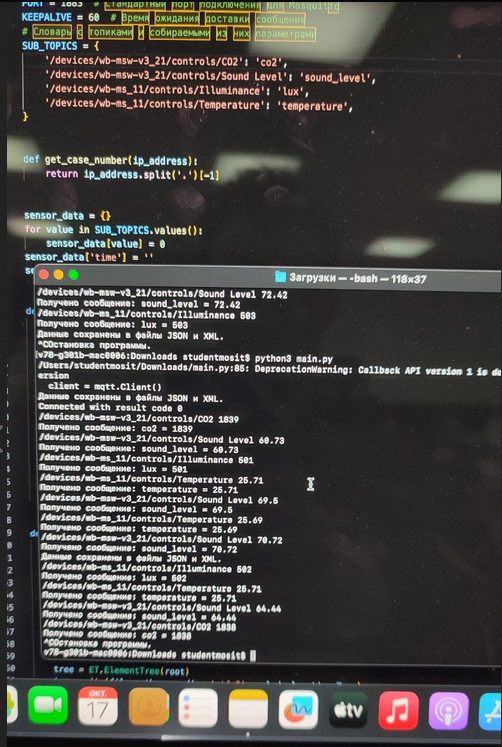


Рисунок 16 – фрагмент программы-парсера для работы с данными из предыдущих пунктов

## Практическая работа №8

* + 1. Постановка задачи

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.

2. Напишите скрипты, позволяющие визуализировать полученные статистические данные.

|  |  |
| --- | --- |
| **№ Варианта** | **Датчики** |
| 2 | 1. Датчик шума устройства WB-MSW v.3 (5)  2. Датчик освещенности устройства WB-MS v.2 (12)  3. Напряжение на любом устройстве стенда |

* + 1. Выполнение задачи

На рисунках 17 - 19 показаны фрагменты файлов, в которых находятся данные, по которым производится построение графиков.

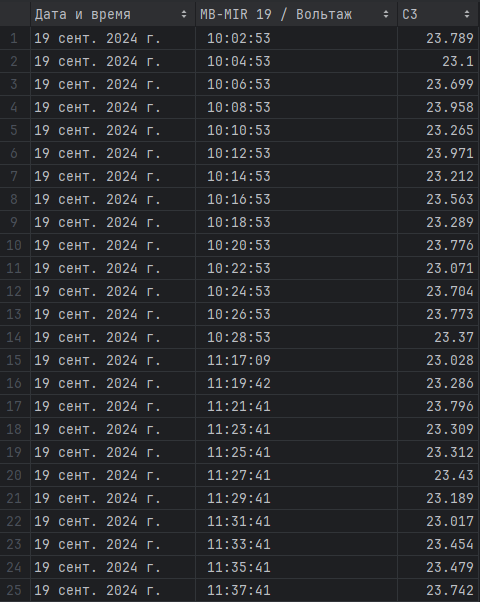


Рисунок 17 – CSV файл с данными вольтажа устройства

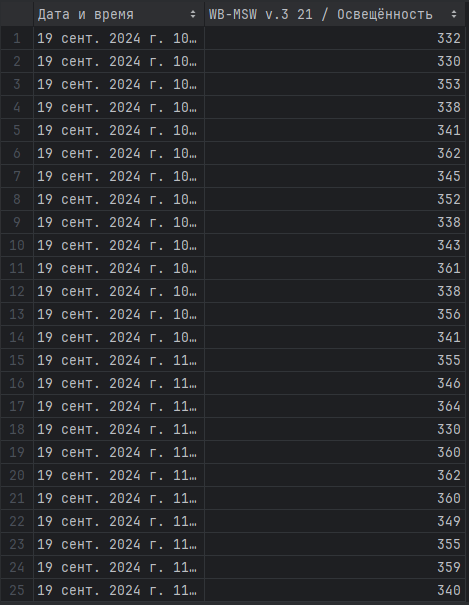


Рисунок 18 – CSV файл с данными освещённости устройства

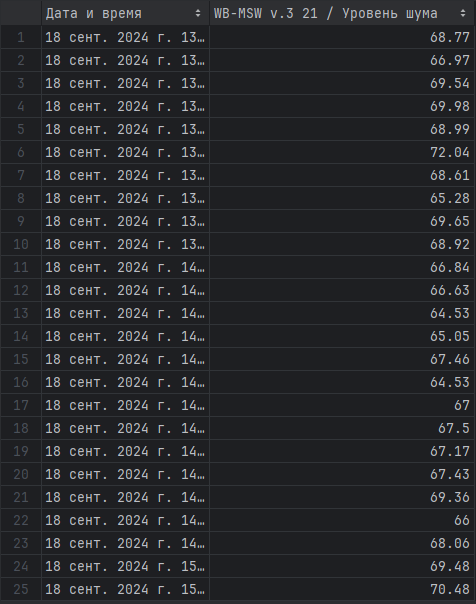


Рисунок 19 – CSV файл с данными уровня шума устройства

Написанные скрипты, позволяющие визуализировать полученные статистические данные в виде: (рисунки 20 - 25).



Рисунок 20 – Код для построения графика шума устройства

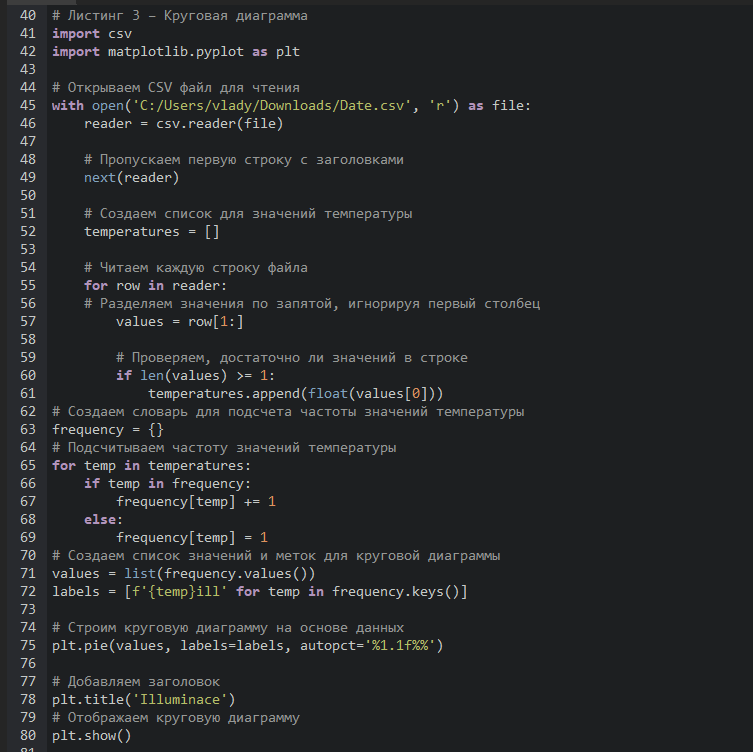


Рисунок 21 – Код для построения графика освещенности

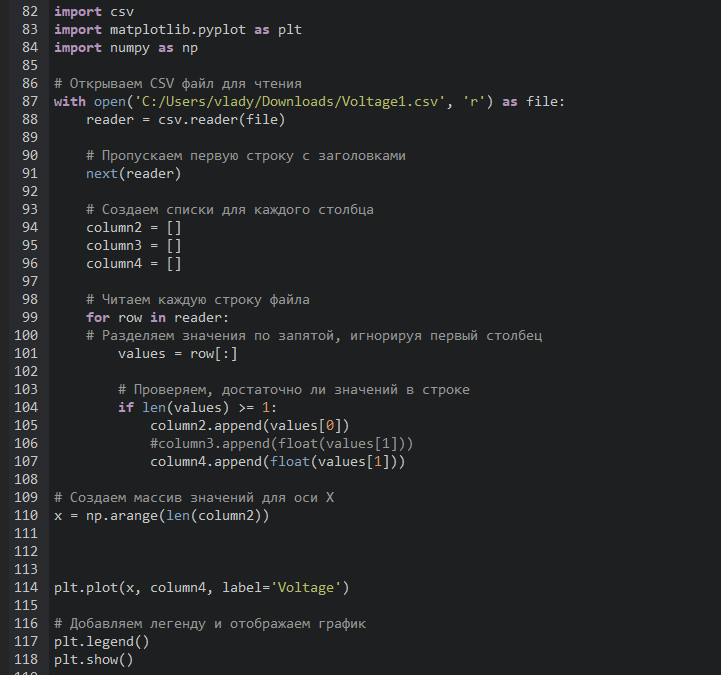


Рисунок 22 – Код для построения графика напряженности

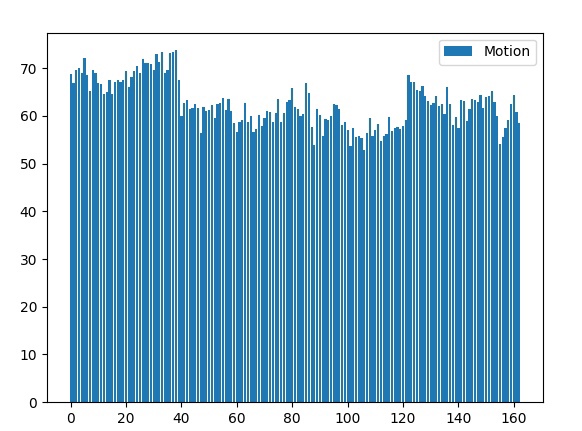


Рисунок 23 - График шума устройства

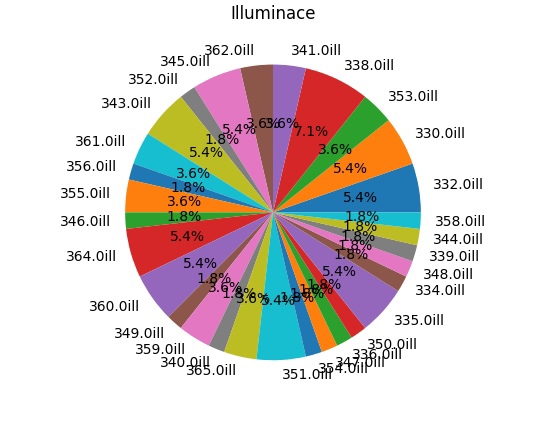


Рисунок 24 – График освещенности устройства

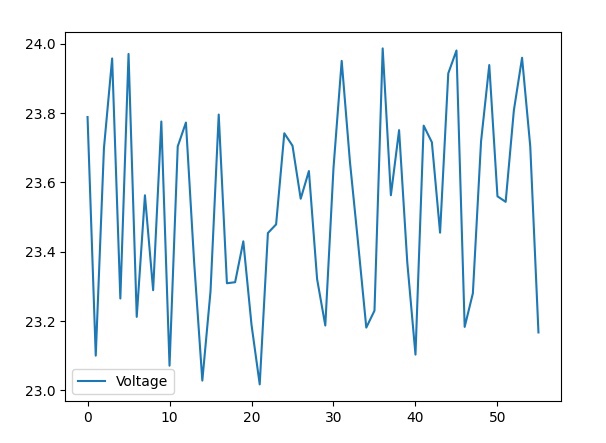


Рисунок 25 - График напряженности устройства

# **Вывод**

В процессе выполнения практических работ мы работыли с демонстрационным стендом WirenBoard и научились использовать MQTT.

# **Список использованных источников**

1. Курс лекций по технологическим основам Интернета Вещей (старший преподаватель кафедры МОСИТ Миронов А. Н.)
2. Документация на чемодан: <https://wirenboard.com/wiki/Wb-demo-kit_v.2>
3. Веб-интерфейс WirenBoard: <https://wirenboard.com/wiki/Wiren_Board_Web_Interface>
4. Полное описание движка правил WirenBoard: https://github.com/wirenboard/wb-rules