

Практическая работа №8 – Визуализация данных в Интернете вещей

Визуализация данных

После сбора показаний с датчиков и сохранения данных в системах интернета вещей зачастую предусмотрена возможность визуализации для пользователей первичных данных или результата их агрегации и анализа.

Визуализация данных — это представление данных в графической форме, обеспечивающей наиболее эффективную работу человека по их изучению, т.е. оценивания информации или данных.

Если проводить классификацию по объектам, то стоит выделить визуализацию:

- числовых данных (детерминированные зависимости — графики, диаграммы, временные ряды; статистические распределения — гистограммы, матрицы диаграмм рассеяния)
- иерархий (диаграммы узлы-связи, дендрограммы)
- сетей (графы, дуговые диаграммы)
- геовизуализацию (карты, картограммы)

В отличие от обычного графического интерфейса, средства визуализации данных обеспечивают способность одновременного отображения большого числа разнотипных данных, возможность их сравнения, выделения выпадающих значений.

Качественная визуализация данных имеет критическое значение для анализа данных и принятия решений на их основе. Продуманное графическое представление не только содержит информацию, но и повышает эффективность ее восприятия за счет наглядности.

Основные способы визуализации (графические представления)

Выбор графического представления осуществляется с учетом типа данных и их предназначения. Некоторые представления лучше подходят для определенного типа данных, чем другие. Большинство средств визуализации предлагает широкий выбор вариантов отображения данных, от обычных линейных графиков и столбчатых диаграмм до временных шкал, карт, зависимостей, гистограмм и настраиваемых представлений.

Графики Линейный график

Линейные графики используются для отображения количественных показателей за непрерывный интервал или определенный период времени. Это

наиболее популярный график для демонстрации трендов и соотношения показателей (при использовании нескольких линий). Линейные графики очень полезны для получения «общей картины» за определенный промежуток времени и наблюдения за развитием в этот период времени.

Чтобы нарисовать линейный график, необходимо сначала отметить точки данных на декартовой системе координат, а затем провести между этими точками линию. Как правило, на оси У отмечаются количественные значения, а на оси X либо качественные значения, либо шкала последовательностей.

Отрицательные значения можно отображать ниже оси X.



Рисунок 31. Структура линейного графика

График плотности

Также известен как график ядра плотности, график плотности распределения вероятности.

График плотности – это инструмент визуализации распределения данных за интервал времени. Этот график представляет собой разновидность гистограммы, где для отображения значений используется ядерное сглаживание, которое позволяет отобразить более гладкое распределение за счет сглаживания изменений параметров. В отличие от гистограмм, они лучше выявляют форму распределения.

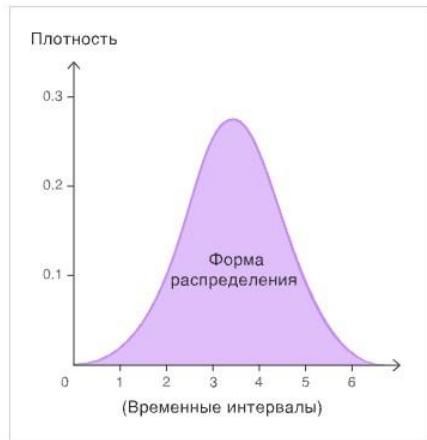


Рисунок 32. Структура графика плотности

62

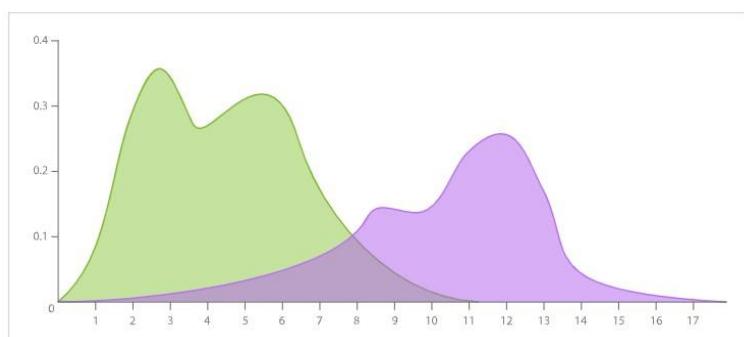


Рисунок 33. Пример графика плотности

Диаграммы

Еще один распространенный способ визуализации. Показывают соотношения набора данных или связи внутри набора данных. В основном строятся вокруг осей, но не всегда. Также их можно построить по секторам или полярной системе координат.

Гистограмма

Гистограмма визуализирует распределение данных в рамках непрерывного интервала или ограниченного периода времени. Каждая полоса на гистограмме представляет в табличной форме частотность за определенный интервал/бин. Общая площадь гистограммы равна количеству данных.

Гистограммы помогают определить концентрацию значений, предельные значения и наличие пробелов или отклонений. Кроме того, они удобны для составления приблизительного обзора распределения вероятностей.

66

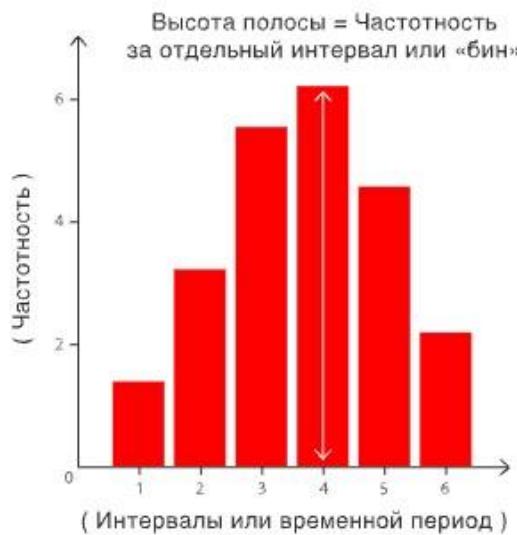


Рисунок 34. Структура гистограммы

Столбиковая диаграмма

Классическая столбиковая диаграмма оперирует горизонтальными или вертикальными столбцами для демонстрации дискретных, числовых сравнений между разными категориями. На одной оси диаграммы представлены конкретные сравниваемые категории, а на другой – шкала дискретных значений.

Столбиковые диаграммы отличаются от гистограмм тем, что не отражают непрерывное развитие в пределах определенного интервала. Дискретные данные столбиковых диаграмм – это данные по категориям, которые отвечают на вопрос: «Сколько?» - по каждой из категорий.

Единственный крупный недостаток столбиковых диаграмм – это оформление условных обозначений при большом количестве полос.



Рисунок 35. Структура столбиковой диаграммы

Столбиковая диаграмма с группировкой

Такой вариант столбиковой диаграммы используется, когда на одной оси располагается два или несколько наборов данных рядом друг с другом и сгруппированных по категориям.

Столбиковые диаграммы с группировкой, как правило, используются для сравнения сгруппированных показателей или категорий с другими группами тех же показателей или типов категории.

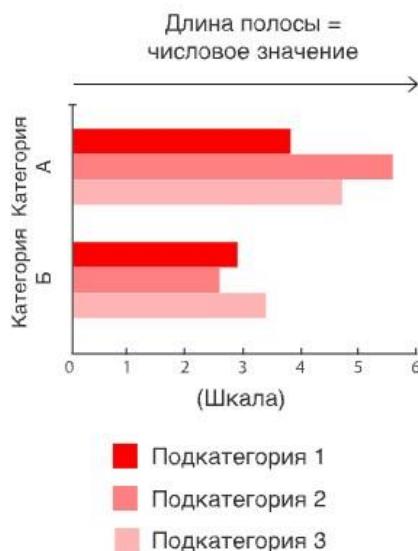


Рисунок 36. Структура столбиковой диаграммы с группировкой

Круговая диаграмма

Известна также как секторная диаграмма в круге.

Круговые диаграммы широко используются в презентациях и офисной документации. Они позволяют показать пропорциональное и процентное соотношение между категориями за счет деления круга на пропорциональные сегменты. Длина каждой дуги представляет собой пропорциональную долю каждой категории, в то время как круг целиком представляет общую сумму всех данных, равную 100%.

Круговые диаграммы идеально подходят для того, чтобы быстро сформировать представление о пропорциональном распределении данных. Однако у этой диаграммы есть ряд существенных недостатков:

Количество отображаемых значений очень ограничено, поскольку, чем больше количество значений, тем меньше размер каждого отдельного сегмента. Соответственно, они не подходят для работы с большими объемами данных.

Они занимают больше места, чем альтернативные графики, например, 100%-ные стопочные диаграммы. Основные причины – размер и, как правило, необходимость в отдельном описании условных обозначений.

Они не очень удобны для проведения точных сравнений между показателями, поскольку визуально площадь сегментов сложнее сравнивать, нежели длину.

Несмотря на это зачастую весьма эффективно сравнивать с помощью круговых диаграмм конкретную категорию (одну часть круга) в рамках общей картины.



Рисунок 37. Структура круговой диаграммы

Радиальная диаграмма

Также известен как паукообразная диаграмма, радарный график, полярный график.

Радиальная диаграмма – это инструмент, позволяющий проводить сравнение между множественными количественными переменными. Именно поэтому он удобен для выявления того, какие переменные имеют одинаковые значения и существуют ли среди значений этих переменных выбросы. Радиальные диаграммы также используются, чтобы выявить максимальные и минимальные значения переменных в пределах набора данных, благодаря чему этот инструмент особенно эффективен для отображения результатов деятельности.

Каждой переменной присваивается ось, имеющая начало в центре. Все оси располагаются радиально с одинаковым промежутком между друг другом, при этом на всех осях используется одна и та же шкала. Сеточные линии, которые соединяют оси между собой, зачастую используются в качестве ориентира. Значение каждой переменной отмечается на ее оси, а все переменные в наборе данных соединяются линиями, образуя многоугольник.

В то же время у радиальной диаграммы есть ряд серьезных недостатков:

При наличии множества многоугольников на одной радиальной диаграмме, она становится трудночитаемой, путанной и перегруженной. Особенно, если многоугольники заполняются цветом, поскольку верхний многоугольник перекрывает собой все остальные.

Слишком большое количество переменных приводит к появлению слишком большого количества осей, что также затрудняет прочтение диаграммы. Соответственно, лучше всего использовать простой вариант графика с ограниченным количеством используемых переменных.

Еще одним недостатком радиальных диаграмм является то, что они не очень удобны для сравнения значений переменных. Несмотря на яркий эффект, создаваемый паутинообразной сеткой, сравнивать значения на одной прямой оси гораздо легче.

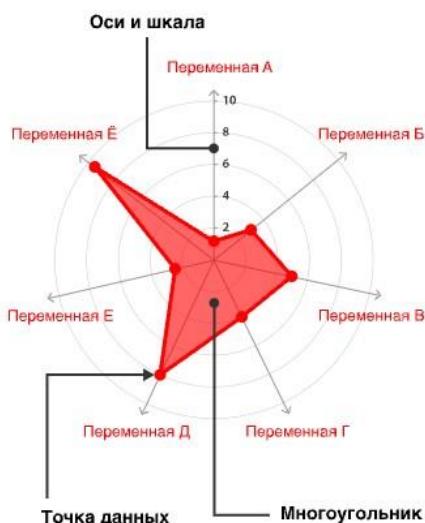


Рисунок 38. Структура радиальной диаграммы

Инструменты визуализации данных

Существует множество инструментов визуализации данных, самые популярные библиотеки для работы с визуализацией в Python:

- Matplotlib <https://matplotlib.org/>

- Seaborn <https://seaborn.pydata.org/>
- Plotly <https://plotly.com/python/>

Каждая из них обладает своими преимуществами и недостатками для использования в конкретных случаях. Далее мы будем рассматривать построение графиков и гистограмм с использованием библиотеки Matplotlib – пожалуй, самой популярной библиотекой Python для визуализации данных.

Линейные графики

Линейные графики являются последовательностью точек данных на линии и самой простой в построении формой визуализации данных. Каждая точка состоит из пары значений (x, y), которые перенесены на график в соответствии с масштабами соответствующих осей.

Листинг 15

Пример построения линейного графика

```
import pandas as pd
import matplotlib.pyplot as plt
```

```
data =
{'series1':[1,3,4,3,5],
'series2':[2,4,5,2,4],
'series3':[3,2,3,1,3]}      df =
pd.DataFrame(data)          x =
np.arange(5)
plt.axis([0,5,0,7])
plt.plot(x,df)
plt.legend(data, loc=2)
plt.show()
```

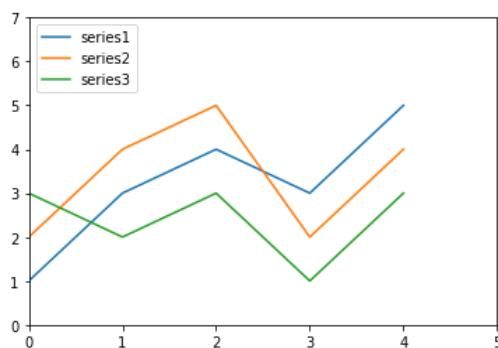


Рисунок 39. Пример построения линейного графика

Гистограммы

Гистограмма состоит из примыкающих прямоугольников, расположенных вдоль оси x, которые разбиты на дискретные интервалы, их называют `bins`. Их площадь пропорциональна частоте конкретного интервала.

Для представления гистограммы в `pyplot` есть функция `hist()`. У нее также есть особенности, которых не найти у других функций, отвечающих за создание графиков. `hist()` не только рисует гистограмму, но также возвращает

67

кортеж значений, представляющих собой результат вычислений гистограммы. Функция `hist()` может реализовывать вычисление гистограммы, чего достаточно для предоставления набора значений и количества интервалов, на которых их нужно разбить. Наконец `hist()` отвечает за разделение интервала на множество и вычисление частоты каждого. Результат этой операции не только выводится в графической форме, но и возвращается в виде кортежа.

Приведем пример. Сгенерируем набор из 100 случайных чисел от 0 до 100 с помощью `random.randint()`.

Листинг 16

Пример генерации набора чисел

```
pop = np.random.randint(0,100,100)

pop
array([33, 90, 10, 68, 18, 67, 6, 54, 32, 25, 90, 6, 48, 34, 59,
70, 37, 50, 86, 7, 49, 40, 54, 94, 95, 20, 83, 59, 33, 0,
81, 18, 26, 69, 2, 42, 51, 7, 42, 90, 94, 63, 14, 14, 71,
25, 85, 99, 40, 62, 29, 42, 27, 98, 30, 89, 21, 78, 17, 33,
63, 80, 61, 50, 79, 38, 96, 8, 85, 19, 76, 32, 19, 14, 37,
62, 24, 30, 19, 80, 55, 5, 94, 74, 85, 59, 65, 17, 80, 11,
81, 84, 81, 46, 82, 66, 46, 78, 29, 40])
```

Создадим гистограмму из этих данных, передавая аргумент функции `hist()`. Разделим данные на 20 интервалов, для этого используется аргумент `bin`:

Листинг 17

Пример построения гистограммы

```
n, bin, patches = plt.hist(pop, bins=20)
plt.show()
```

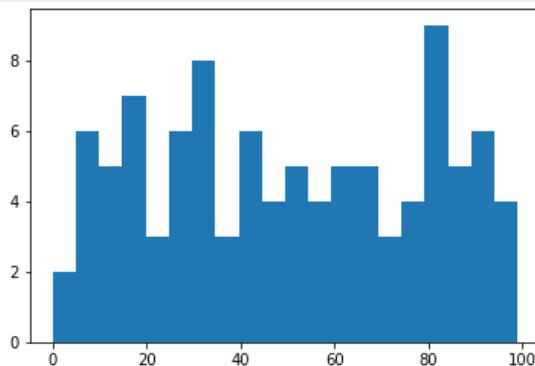


Рисунок 40. Пример построения гистограммы

Круговая диаграмма

Еще один способ представления данных — круговая диаграмма, которую можно получить с помощью функции `pie()`.

68

Даже для нее нужно передать основной аргумент, представляющий собой список значений. Пусть это будут проценты (где максимально значение — 100), но это может быть любое значение. А уже сама функция определит, сколько будет занимать каждое значение.

Если нужно задать последовательность цветов, используется аргумент `colors`. В таком случае придется присвоить список строк, каждая из которых будет содержать название цвета. Также можно добавить метки для каждой доли, используя аргумент `labels`, которой присваивает список строк с метками в последовательности.

Чтобы диаграмма была идеально круглой, необходимо в конце добавить функцию `axis()` со строкой `equal` в качестве аргумента.

Листинг 18

Пример 1 построения круговой диаграммы

```
labels = ['Nokia', 'Samsung', 'Apple', 'Lumia']
values = [10, 30, 45, 15]
colors = ['yellow', 'green', 'red', 'blue']
plt.pie(values, labels=labels, colors=colors)
plt.axis('equal')
plt.show()
```

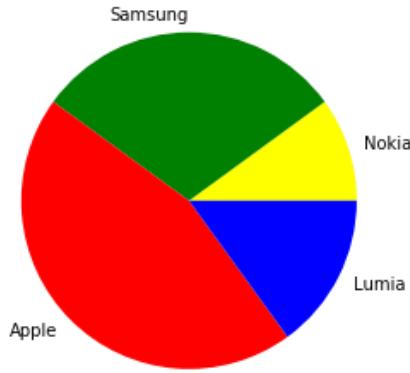


Рисунок 41. Пример построения круговой диаграммы

Аргумент `explode` используется для того, чтобы «вытащить» одну из секций. Он представляет собой всего лишь последовательность чисел с плавающей точкой от 0 до 1, где 1 — положение целиком вне диаграммы, а 0 — полностью внутри. Значение между соответствуют среднему градусу извлечения.

С помощью функции `title()` можно добавить заголовок к диаграмме. Также можно настроить угол поворота с помощью аргумента `startangle`, который принимает значение между 0 и 360, обозначающее угол поворота (0 — значение по умолчанию). Следующий график показывает все изменения.

Листинг 19

Пример 2 построения круговой диаграммы

```

labels = ['Nokia', 'Samsung', 'Apple', 'Lumia']
values = [10, 30, 45, 15]
colors = ['yellow', 'green', 'red', 'blue']
explode = [0.3, 0, 0, 0]
plt.title('A Pie Chart')
plt.pie(values, labels=labels, colors=colors, explode=explode, startangle=180)
plt.axis('equal')
plt.show()

```

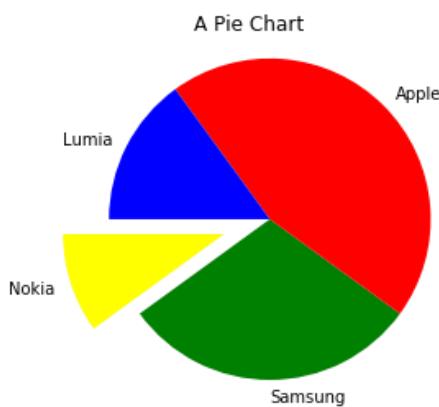


Рисунок 42. Пример построения круговой диаграммы

Аргумент `autopct`, добавляет в центр каждой части текст со значением доли в процентном соотношении.

Листинг 20

Пример 3 построения круговой диаграммы

```

labels =
['Nokia', 'Samsung', 'Apple', 'Lumia']      values =
[10, 30, 45, 15]      colors =
['yellow', 'green', 'red', 'blue']      explode =
[0.3, 0, 0, 0]      plt.title('A Pie Chart')

      plt.pie(values, labels=labels, colors=colors, explode=explode, shadow
= True, autopct='%1.1f%%', startangle=180)      plt.axis('equal')
      plt.show()

```

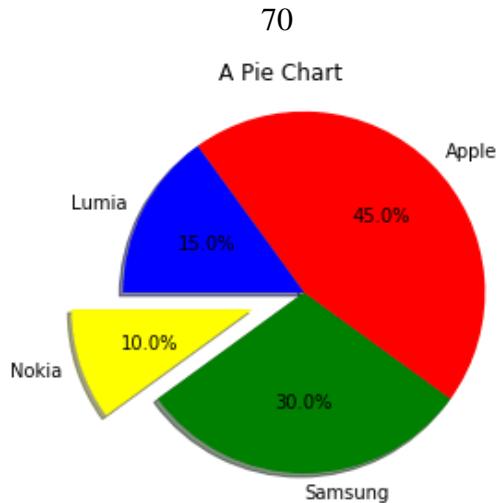


Рисунок 43. Пример построения круговой диаграммы

Задание практической работы №8 Часть 1.

Подпишитесь на несколько MQTT топиков в составе стенда **WB-demo-kit v.2**

1. согласно вариантам с компьютера в аудитории или личного устройства и собираите данные с датчиков в течение 10 или более минут.

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

Таблица 9. Варианты для выполнения практической работы № 8

№ варианта	Датчики
1	<ol style="list-style-type: none"> 1. Датчик температуры устройства WB-MSW v.3 (5) 2. Датчик движения устройства WB-MSW v.3 (5) 3. Напряжение на любом устройстве стенда
2	<ol style="list-style-type: none"> 1. Датчик шума устройства WB-MSW v.3 (5) 2. Датчик освещенности устройства WB-MS v.2 (12) 3. Напряжение на любом устройстве стенда
3	<ol style="list-style-type: none"> 1. Датчик шума устройства WB-MSW v.3 (5) 2. Датчик CO2 устройства WB-MSW v.3 (5) 3. Напряжение на любом устройстве стенда

4	1. Датчик движения устройства WB-MSW v.3 (5) 2. Датчик температуры устройства WB-MS v.2 (12) 3. Напряжение на любом устройстве стенда
5	1. Датчик CO2 устройства WB-MSW v.3 (5) 2. Датчик освещенности устройства WB-MS v.2 (12) 3. Напряжение на любом устройстве стенда
6	1. Датчик влажности устройства WB-MSW v.3 (5) 2. Датчик шума устройства WB-MSW v.3 (5) 3. Напряжение на любом устройстве стенда

Окончание табл. 9

7	1. Датчик влажности устройства WB-MSW v.3 (5) 2. Датчик температуры 1-wire DS18B20 (1) 3. Напряжение на любом устройстве стенда
---	---

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

- Столбиковой диаграммы (гистограммы) (например, частоты показаний датчика)



Рисунок 44. Пример построения столбиковой диаграммы

- Линейного графика (например, показаний по времени)



Рисунок 45. Пример построения линейного графика

- Круговой диаграммы (например, распределения показаний по времени)

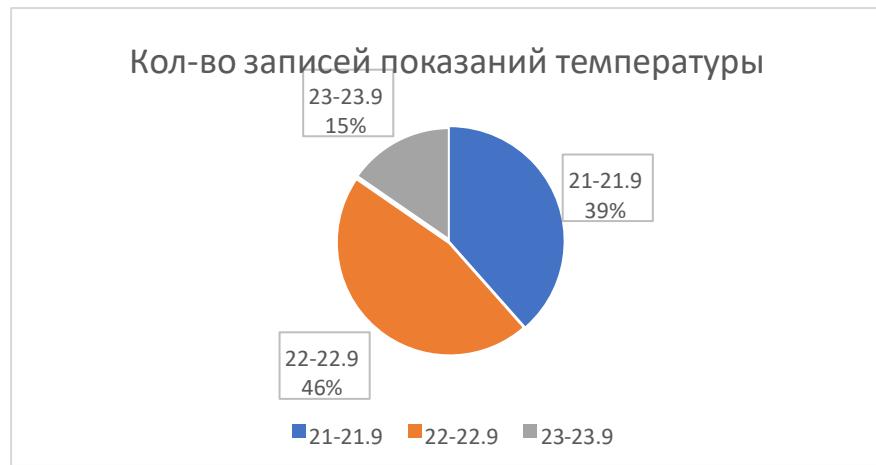


Рисунок 46. Пример построения круговой диаграммы

Необходимо реализовать визуализацию данных всех трех перечисленных типов (т.е. для каждого датчика необходимо выбрать тип визуализации на свое усмотрение и не повторяться в типах визуализации в рамках одного варианта).

Задание практической работы №8 Часть 2.

Подпишитесь на несколько MQTT топиков **в составе стенда WB-demo-kit v.3**

1. согласно вариантам с компьютера в аудитории или личного устройства и собираите данные с датчиков в течение 10 или более минут.

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

Таблица 10. Варианты для выполнения практической работы № 8

№ варианта	Датчики
1	<ol style="list-style-type: none"> 1. Датчик температуры устройства 1-wire DS18B20 (BK1) 2. Датчик VOC устройства WB-MSW v.4 (A6) 3. Напряжение на любом устройстве стенда
2	<ol style="list-style-type: none"> 1. Датчик уровня шума устройства WB-MSW v.4 (A6) 2. Датчик освещенности устройства WB-MS v.4 (A6) 3. Напряжение на любом устройстве стенда

3	1. Датчик температуры устройства 1-wire DS18B20 (ВК2) 2. Датчик CO2 устройства WB-MSW v.4 (A6) 3. Напряжение на любом устройстве стенда
4	1. Датчик движения устройства WB-MSW v.4 (A6) 2. Датчик температуры устройства WB-MS v.4 (A6) 3. Напряжение на любом устройстве стенда
5	1. Датчик CO2 устройства WB-MSW v.4 (A6) 2. Датчик освещенности устройства WB-MS v.4 (A6) 3. Напряжение на любом устройстве стенда
6	1. Датчик влажности устройства WB-MSW v.4 (A6) 2. Датчик движения устройства WB-MSW v.4 (A6) 3. Напряжение на любом устройстве стенда
7	1. Датчик освещенности устройства WB-MS v.4 (A6) 2. Датчик VOC устройства WB-MSW v.4 (A6) 3. Напряжение на любом устройстве стенда

Далее, как и в части 1, необходимо реализовать визуализацию данных всех трех перечисленных типов (т.е. для каждого датчика необходимо выбрать тип визуализации на свое усмотрение и не повторяться в типах визуализации в рамках одного варианта).

Дополнительное задание практической работы №8

Поднимите на физическом устройстве (или на устройстве, где запущен его программный эмулятор) MQTT-брюкер. Показания с датчиков должны публиковаться в соответствующие им топики.

Подпишитесь с другого устройства (компьютера в аудитории или личного устройства) на топики физического устройства. Полученные от MQTT-брюкера данные сохраняйте в локальную базу данных.

В отчете опишите процесс настройки MQTT-брюкера (снабжая текстовое описание скриншотами), структуру топиков, процесс подключения другого устройства к MQTT-брюкеру и результат получения и сохранения данных от физического устройства.

Требования к отчету по блоку практических работ №2:

По итогу выполнения практических работ №5-8, необходимо оформить единый отчёт, включающий:

1. Титульный лист;
2. Оглавление;
3. Описание датчиков и актуаторов, технологий передачи данных из основной части ПР №5;
4. Процесс и результат выполнения (текстовое описание выполненных действий, команды и результат их выполнения в виде скриншотов) основных заданий ПР №6;
5. Листинги с комментариями и результат выполнения работы в виде скриншотов (например, сгенерированных файлов) из основной части ПР №7;
6. Листинги и полученные графики, диаграммы и т.д. из основной части ПР №8;
7. Отчет о проекте – результаты выполнения всех дополнительных заданий ПР №5-8; 8. Выводы о проделанной работе.

Отчет по практическим работам необходимо загрузить в СДО (в случае каких-либо технических проблем отчет необходимо выслать на почту преподавателя в домене *mirea.ru*)