



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**«МИРЭА – Российский технологический университет»**  
**РГУ МИРЭА**

---

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных  
технологий

## **Отчет по практическим работам №9-12**

по дисциплине «Технологические основы Интернета вещей»

**Выполнили:**

Студенты группы ИКБО-20-23

Кузнецов Л.А.  
Комисарик М.А.

**Проверил:**

Жматов Дмитрий Владимирович

2025 г.

## **ОГЛАВЛЕНИЕ**

1. Практическая работа №9 .....	3
2. Практическая работа №10 .....	10
3. Практическая работа №11 .....	13
4. Практическая работа №12 .....	17
ВЫВОД.....	20

# ПРАКТИЧЕСКАЯ ЧАСТЬ

## 1. Практическая работа №9

### Часть 1. Регистрация на платформе Rightech IoT Cloud

Для выполнения работ была осуществлена регистрация на сайте [dev.rightech.io](http://dev.rightech.io).

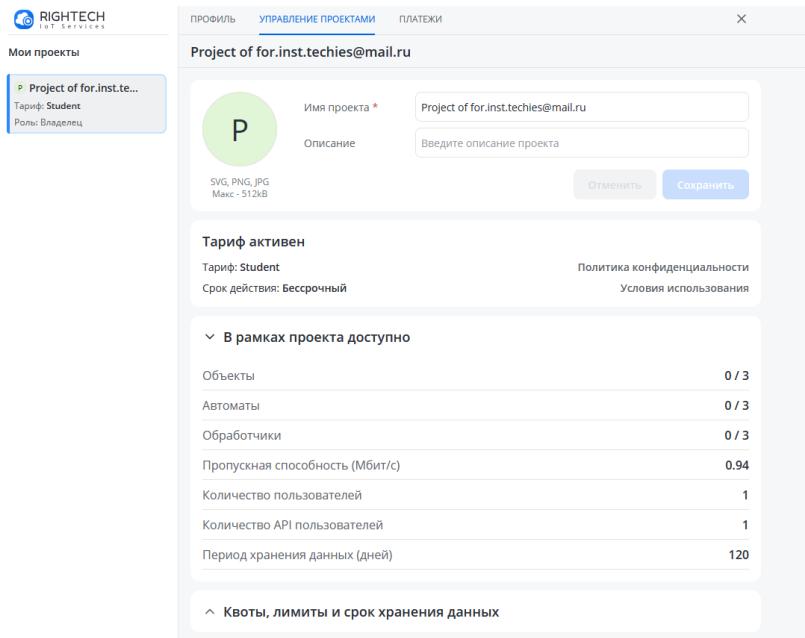


Рисунок 1.1.1 – Вид профиля зарегистрированного пользователя на [dev.rightech.io](http://dev.rightech.io)

### Часть 2. Создание виртуальных устройств в облаке

В облаке были созданы виртуальные устройства согласно варианту.

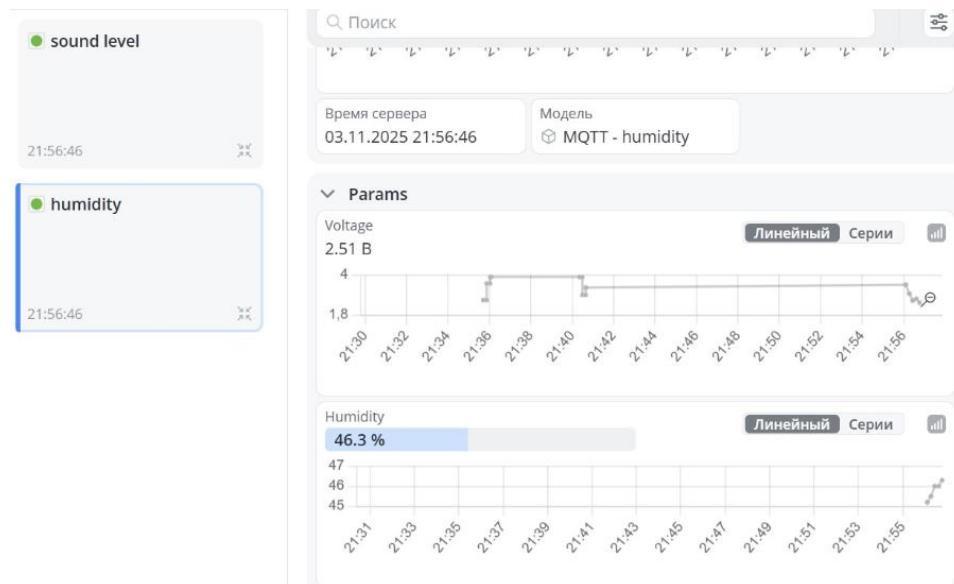


Рисунок 1.2.1 – Показатель напряжения на датчик влажности

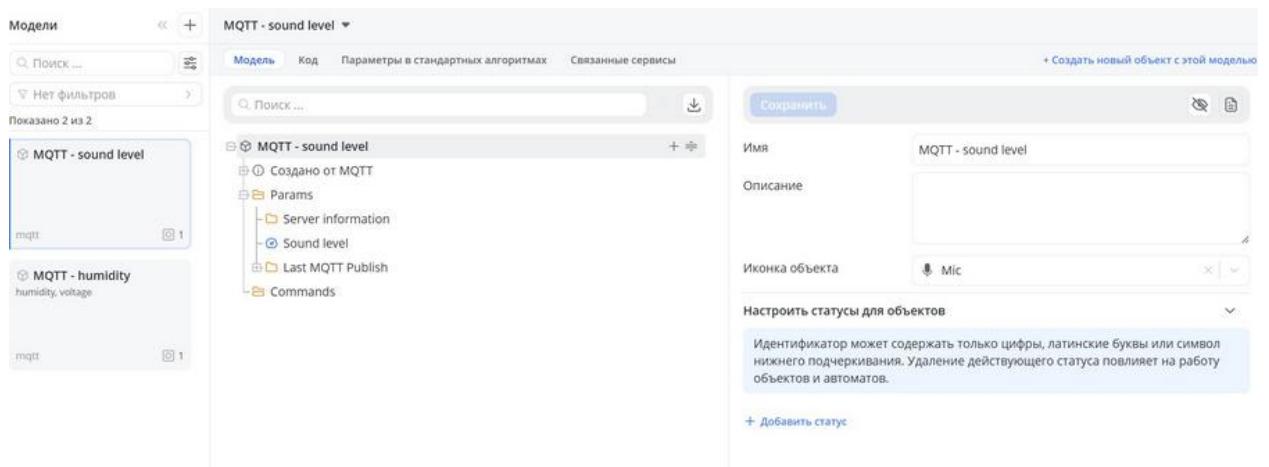


Рисунок 1.2.2 – Модель датчика громкости звука часть 1

**Сохранить**

Тип	<b>Аргумент</b>
Идентификатор	sound_level
Имя	Sound level
Описание	
Тип данных	number : Число
Источник	/devices/sensors/s2/sound level
Источник данных зависит от протокола	
Линейный	<input checked="" type="checkbox"/>
<b>При разборе</b>	
<b>Отображение</b>	
Настройка отображения параметра в меню Объект	
Единица измерения	декибелы (дБ)
Количество знаков после запятой	2

Рисунок 1.2.3 – Модель датчика громкости звука часть 2

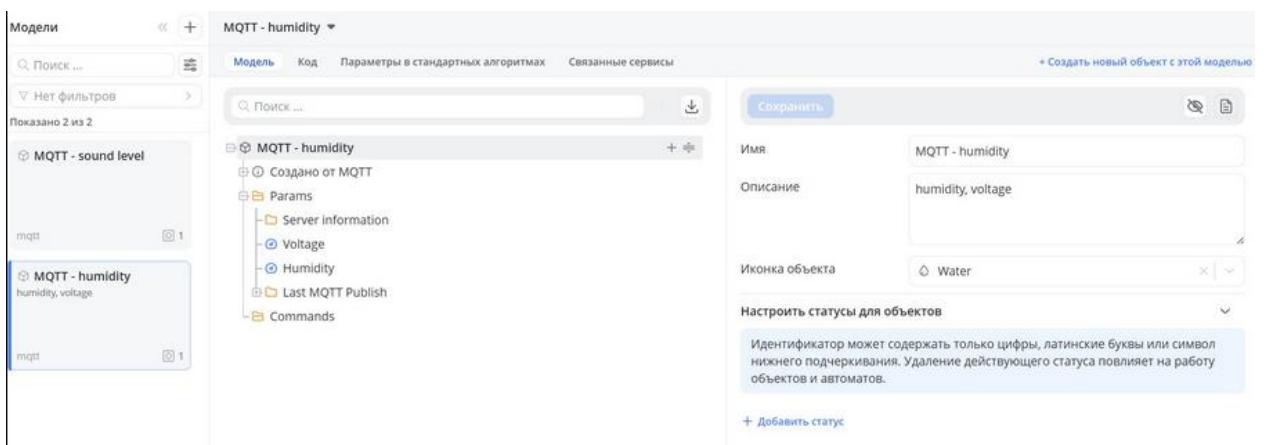


Рисунок 1.2.4 – Модель датчика влажности часть 1

Тип	 Аргумент
Идентификатор	humidity
Имя	Humidity
Описание	
Тип данных	number : Число
Источник	/devices/sensors/s1/humidity Источник данных зависит от протокола
Линейный	<input checked="" type="checkbox"/>
При разборе	
Отображение	
Настройка отображения параметра в меню Объект	
Единица измерения	проценты (%)
Количество знаков после запятой	1

Рисунок 1.2.5 – Модель датчика влажности часть 2

Также с моделями были созданы объекты датчиков.

The screenshot shows the Rightech interface for creating a new object. On the left, a sidebar lists objects: 'sound level' (selected) and 'humidity'. The main panel is titled 'sound level'. It contains fields for 'Модель\*' (Model\*) set to 'MQTT - sound level', 'Идентификатор\*' (Identifier\*) set to 'mqtt-mihakom050-s2', 'Имя\*' (Name\*) set to 'sound level', and 'Описание' (Description) with the note 'произвольное, отображается в интерфейсе пользователя' (arbitrary, displayed in the user interface). There are also buttons for 'Сохранить' (Save), 'Журнал' (Journal), 'Редактирование' (Edit), 'Команды' (Commands), and 'Сервисы' (Services).

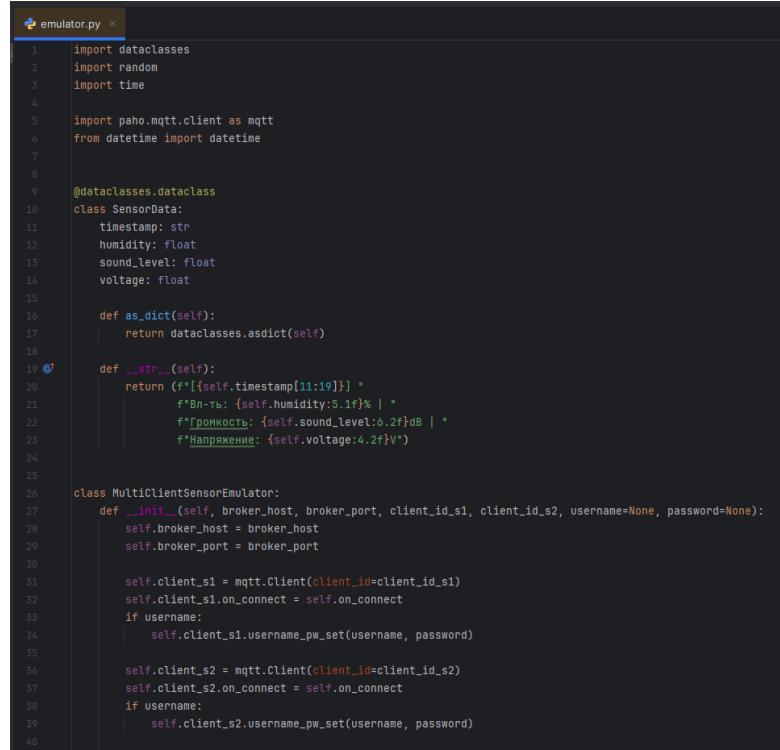
Рисунок 1.2.6 – Объект датчика громкости (шума)

The screenshot shows the Rightech interface for creating a new object. On the left, a sidebar lists objects: 'sound level' (selected) and 'humidity'. The main panel is titled 'humidity'. It contains fields for 'Модель\*' (Model\*) set to 'MQTT - humidity', 'Идентификатор\*' (Identifier\*) set to 'mqtt-mihakom050-s1', 'Имя\*' (Name\*) set to 'humidity', and 'Описание' (Description) with the note 'произвольное, отображается в интерфейсе пользователя' (arbitrary, displayed in the user interface). There are also buttons for 'Сохранить' (Save), 'Журнал' (Journal), 'Редактирование' (Edit), 'Команды' (Commands), and 'Сервисы' (Services).

Рисунок 1.2.7 – Объект датчика влажности

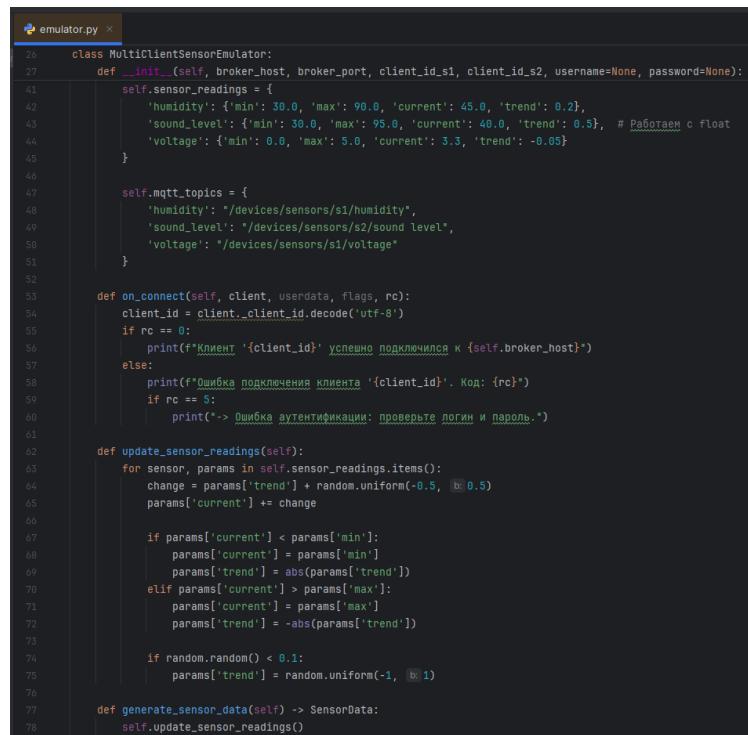
## Часть 3. Отправка данных в облако

Отправка в облако проводилась при помощи эмулятора написанного на языке Python.



```
emulator.py
1 import dataclasses
2 import random
3 import time
4
5 import paho.mqtt.client as mqtt
6 from datetime import datetime
7
8
9 @dataclasses.dataclass
10 class SensorData:
11     timestamp: str
12     humidity: float
13     sound_level: float
14     voltage: float
15
16     def as_dict(self):
17         return dataclasses.asdict(self)
18
19     def __str__(self):
20         return f'{self.timestamp[11:19]} | '
21         f'Вл-ть: {self.humidity:.5.1f}% | '
22         f'Громкость: {self.sound_level:.2f}dB | '
23         f'Напряжение: {self.voltage:.4.2f}V'
24
25
26 class MultiClientSensorEmulator:
27     def __init__(self, broker_host, broker_port, client_id_s1, client_id_s2, username=None, password=None):
28         self.broker_host = broker_host
29         self.broker_port = broker_port
30
31         self.client_s1 = mqtt.Client(client_id=client_id_s1)
32         self.client_s1.on_connect = self.on_connect
33         if username:
34             self.client_s1.username_pw_set(username, password)
35
36         self.client_s2 = mqtt.Client(client_id=client_id_s2)
37         self.client_s2.on_connect = self.on_connect
38         if username:
39             self.client_s2.username_pw_set(username, password)
40
```

Рисунок 1.3.1 – Код эмулятора с подключением к брокеру по адресу dev.rigtech.io часть 1



```
emulator.py
20     class MultiClientSensorEmulator:
21         def __init__(self, broker_host, broker_port, client_id_s1, client_id_s2, username=None, password=None):
22             self.sensor_readings = {
23                 'humidity': {'min': 30.0, 'max': 90.0, 'current': 45.0, 'trend': 0.2},
24                 'sound_level': {'min': 30.0, 'max': 95.0, 'current': 40.0, 'trend': 0.5}, # Работаем с float
25                 'voltage': {'min': 0.0, 'max': 5.0, 'current': 3.3, 'trend': -0.05}
26             }
27
28             self.mqtt_topics = {
29                 'humidity': '/devices/sensors/s1/humidity',
30                 'sound_level': '/devices/sensors/s2/sound level',
31                 'voltage': '/devices/sensors/s1/voltage'
32             }
33
34         def on_connect(self, client, userdata, flags, rc):
35             client_id = client._client_id.decode("utf-8")
36             if rc == 0:
37                 print(f"Клиент '{client_id}' успешно подключился к {self.broker_host}")
38             else:
39                 print(f"Ошибка подключения клиента '{client_id}'. Код: {rc}")
40                 if rc == 5:
41                     print("=> Ошибка аутентификации: проверьте логин и пароль.")
42
43         def update_sensor_readings(self):
44             for sensor, params in self.sensor_readings.items():
45                 change = params['trend'] + random.uniform(-0.5, 0.5)
46                 params['current'] += change
47
48                 if params['current'] < params['min']:
49                     params['current'] = params['min']
50                     params['trend'] = abs(params['trend'])
51                 elif params['current'] > params['max']:
52                     params['current'] = params['max']
53                     params['trend'] = -abs(params['trend'])
54
55                 if random.random() < 0.1:
56                     params['trend'] = random.uniform(-1, 1)
57
58         def generate_sensor_data(self) -> SensorData:
59             self.update_sensor_readings()
60
```

Рисунок 1.3.2 – Код эмулятора с подключением к брокеру по адресу dev.rigtech.io часть 2

```

16     class MultiClientSensorEmulator:
17         def generate_sensor_data(self) -> SensorData:
18             ...
19             return SensorData(
20                 timestamp=datetime.now().isoformat(),
21                 humidity=round(self.sensor_readings['humidity']['current'], 1),
22                 sound_level=round(self.sensor_readings['sound_level']['current'], 2),
23                 voltage=round(self.sensor_readings['voltage']['current'], 2)
24             )
25
26         def publish(self, sensor_data: SensorData):
27             self.client_s1.publish(self.mqtt_topics['humidity'], sensor_data.humidity)
28             self.client_s1.publish(self.mqtt_topics['voltage'], sensor_data.voltage)
29
30             self.client_s2.publish(self.mqtt_topics['sound_level'], sensor_data.sound_level)
31
32             print(sensor_data)
33
34         def start_emulation(self, interval):
35             print("Запуск эмуляции датчиков...")
36             try:
37                 self.client_s1.connect(self.broker_host, self.broker_port, keepalive=60)
38                 self.client_s2.connect(self.broker_host, self.broker_port, keepalive=60)
39
40                 self.client_s1.loop_start()
41                 self.client_s2.loop_start()
42
43                 while True:
44                     data = self.generate_sensor_data()
45                     self.publish(data)
46                     time.sleep(interval)
47             except KeyboardInterrupt:
48                 print("\n\nЭмуляция остановлена пользователем.")
49             except Exception as e:
50                 print(f"Произошла ошибка: {e}")
51             finally:
52                 self.client_s1.loop_stop()
53                 self.client_s1.disconnect()
54                 self.client_s2.loop_stop()
55                 self.client_s2.disconnect()
56                 print("Соединения с MQTT брокером закрыты.")
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129

```

Рисунок 1.3.3 – Код эмулятора с подключением к брокеру по адресу dev.rigtech.io часть 3

```

109
110
111
112
113
114
115
116
117
118
119
120
121 if __name__ == "__main__":
122     emulator = MultiClientSensorEmulator(
123         broker_host="dev.rigtech.io",
124         broker_port=1883,
125         client_id_s1="mqtt-mihakom050-s1",
126         client_id_s2="mqtt-mihakom050-s2"
127     )
128     emulator.start_emulation(interval=10)
129

```

Рисунок 1.3.4 – Код эмулятора с подключением к брокеру по адресу dev.rigtech.io часть 4

```

1   import dataclasses
2   import random
3   import time
4
5   import paho.mqtt.client as mqtt
6   from datetime import datetime
7
8
9   @dataclasses.dataclass
10  class SensorData:
11      timestamp: str
12      humidity: float
13      sound_level: float
14      voltage: float
15
16      def __dict__(self):
17          return dataclasses.asdict(self)
18
19      def __str__(self):
20          return f'{self.timestamp} | ' +
21                 f'Вл-ть: {self.humidity} | ' +
22                 f'Громкость: {self.sound_level}dB | ' +
23                 f'Напряжение: {self.voltage}V'
24
25
26  class MultiClientSensorEmulator:
27
28      def __init__(self):
29          ...
30
31          self.client_s1 = mqtt.Client(client_id=self.client_id_s1)
32          self.client_s2 = mqtt.Client(client_id=self.client_id_s2)
33
34          self.client_s1.on_connect = self._on_connect
35          self.client_s2.on_connect = self._on_connect
36
37          self.client_s1.on_message = self._on_message
38          self.client_s2.on_message = self._on_message
39
40          self.client_s1.on_disconnect = self._on_disconnect
41          self.client_s2.on_disconnect = self._on_disconnect
42
43          self.client_s1.on_subscribe = self._on_subscribe
44          self.client_s2.on_subscribe = self._on_subscribe
45
46          self.client_s1.on_unsubscribe = self._on_unsubscribe
47          self.client_s2.on_unsubscribe = self._on_unsubscribe
48
49          self.client_s1.on_log = self._on_log
50          self.client_s2.on_log = self._on_log
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129

```

Рисунок 1.3.5 – Код эмулятора с подключением к брокеру по адресу dev.rigtech.io часть 5

Корректность работы эмулятора была проверена в IDE, а также на сайте [dev.rightech.io](http://dev.rightech.io).

```
C:\Users\Noip\Downloads\emulator.py:31: DeprecationWarning: Callback API version 1 is deprecated, update to latest version
    self.client_s1 = mqtt.Client(client_id=client_id_s1)
C:\Users\Noip\Downloads\emulator.py:36: DeprecationWarning: Callback API version 1 is deprecated, update to latest version
    self.client_s2 = mqtt.Client(client_id=client_id_s2)
Запуск эмуляции датчиков...
[12:39:24] Вл-ть: 45.2% | Громкость: 40.66dB | Напряжение: 2.98V
Клиент 'mqtt-mihakom@50-s1' успешно подключился к dev.rightech.io
Клиент 'mqtt-mihakom@50-s2' успешно подключился к dev.rightech.io
[12:39:34] Вл-ть: 45.6% | Громкость: 41.26dB | Напряжение: 3.11V
[12:39:44] Вл-ть: 45.4% | Громкость: 42.34dB | Напряжение: 2.91V
```

Рисунок 1.3.6 – Итоговый вид результата работы эмулятора

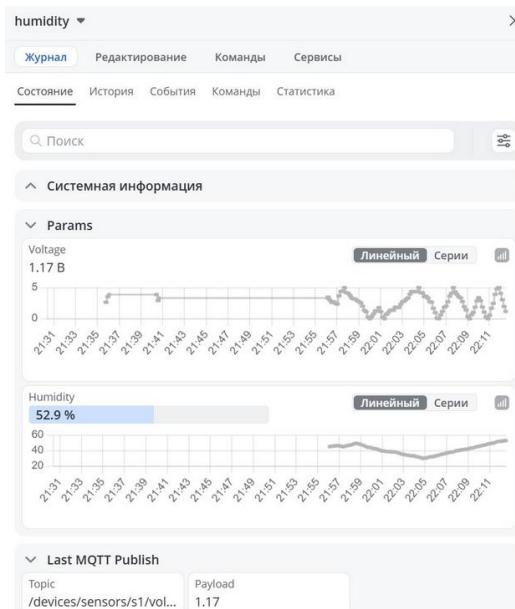


Рисунок 1.3.7 – Получение данных влажности и напряжения

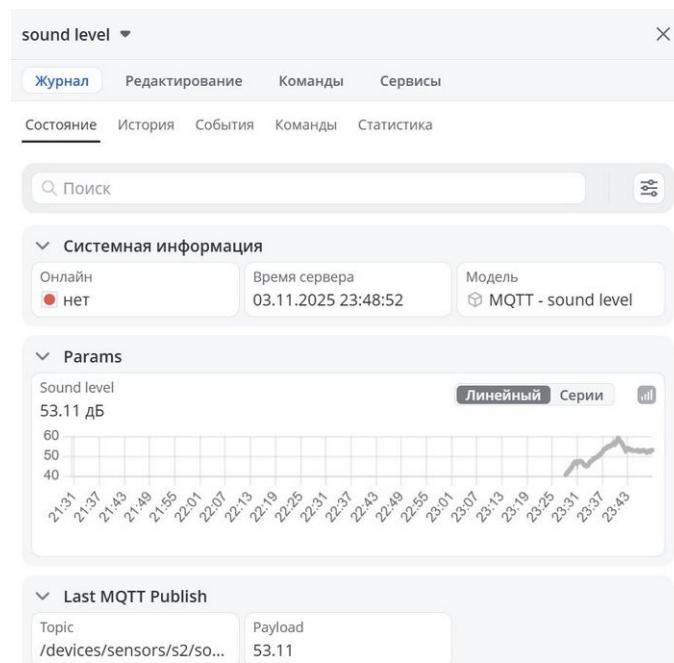


Рисунок 1.3.8 – Получение данных громкости звука

```

Запуск эмуляции датчиков...
[23:30:33] Вл-ть: 44.0% | Громкость: 47.29dB | Напряжение: 5.00V
[23:30:44] Вл-ть: 44.2% | Громкость: 47.31dB | Напряжение: 4.25V
[23:30:54] Вл-ть: 44.6% | Громкость: 46.22dB | Напряжение: 3.88V
[23:31:04] Вл-ть: 45.5% | Громкость: 46.52dB | Напряжение: 3.27V
[23:31:14] Вл-ть: 46.1% | Громкость: 47.27dB | Напряжение: 3.00V
[23:31:24] Вл-ть: 46.9% | Громкость: 47.41dB | Напряжение: 3.15V
[23:31:34] Вл-ть: 47.8% | Громкость: 47.63dB | Напряжение: 3.80V
[23:31:44] Вл-ть: 48.4% | Громкость: 47.59dB | Напряжение: 4.09V
[23:31:54] Вл-ть: 47.9% | Громкость: 47.50dB | Напряжение: 3.87V
[23:32:04] Вл-ть: 48.0% | Громкость: 47.14dB | Напряжение: 4.19V
[23:32:14] Вл-ть: 48.2% | Громкость: 47.03dB | Напряжение: 4.18V

```

Рисунок 1.3.9 – Отчёт об отправке данных эмулятором

На рисунках 1.3.7-1.3.8 представлены полученные облаком данные, а на рисунках 1.3.6 и 1.3.9 – процесс проверки корректности работы кода эмулятора.

## Часть 2

### 2. Практическая работа №10

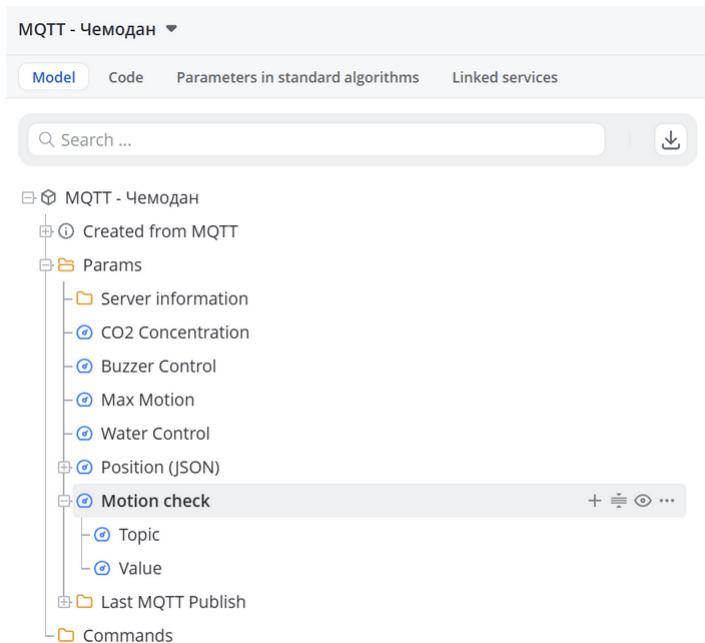


Рисунок 2.1 – Настройка моделей

По варианту необходимо воспроизвести сценарии:

- Включения и выключения звукового сигнала по концентрации CO2;

- Включения и выключения воды по датчику движения.

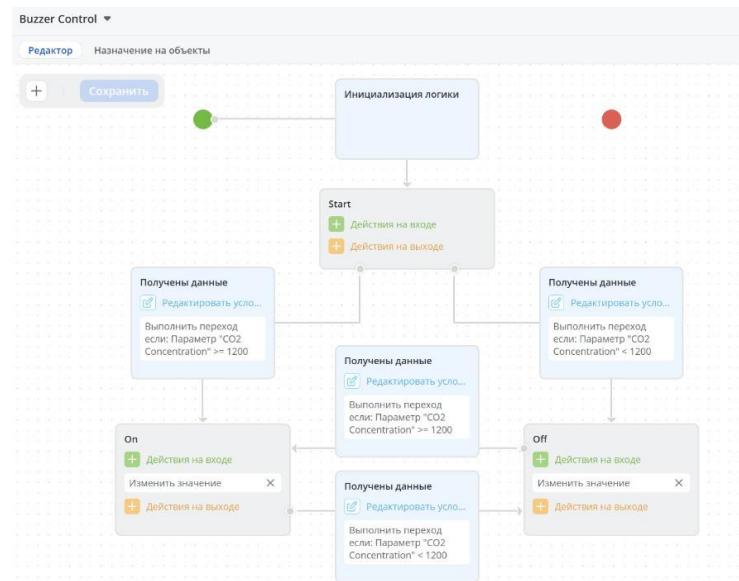


Рисунок 2.2 – Воспроизведение сценария включения и выключения звукового сигнала по концентрации CO2



Рисунок 2.3 – Результаты тестирования воспроизведения сценария включения и выключения звукового сигнала по концентрации CO2

На рисунках 2.2 – 2.3 происходит подключение к получению данных с датчика с CO2, после чего в зависимости от изменения значений датчика происходит включение (при концентрации CO2 большей 1200) и выключение (при концентрации CO2 не меньшей 1200) звукового сигнала.

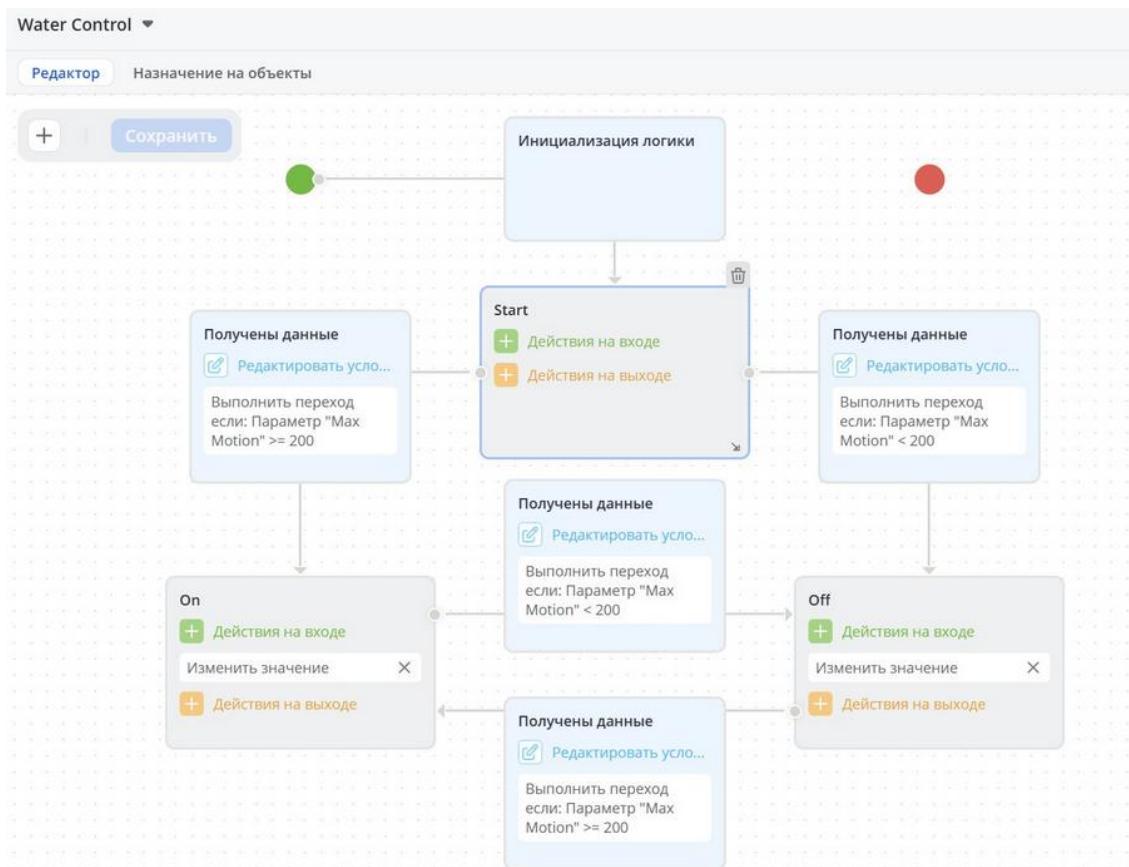


Рисунок 2.4 – Воспроизведение сценария включения и выключения воды по датчику движения

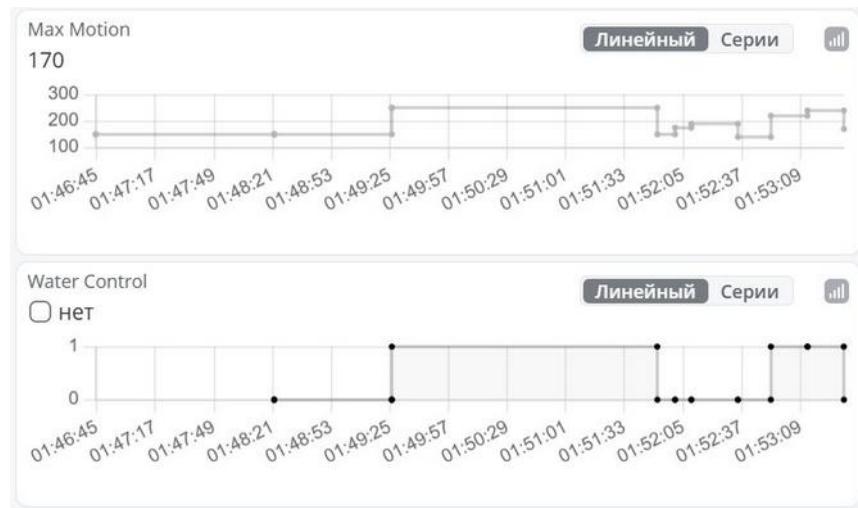


Рисунок 2.5 – Результаты тестирования воспроизведения сценария включения и выключения воды по датчику движения

На рисунках 2.4 – 2.5 происходит подключение к получению данных с датчика движения, после чего в зависимости от изменения значений датчика происходит включение (при зафиксированном перемещении не меньше 200 интенсивности движения) и выключение (при зафиксированном перемещении меньше 200 интенсивности движения) воды.

```

20bridges.conf ×
1 #rightech connect config
2 connection rightech
3 address dev.rightech.io:1883
4 clientid mqtt-mihakom050-case
5 try_private false
6 start_type automatic
7 topic /devices/wb-msw-v3_64/controls/CO2 both
8 topic /devices/wb-msw-v3_64/controls/Buzzer/on both
9 topic /devices/wb-msw-v3_64/controls/Max Motion both
10 topic /devices/wb-mwac-v2_25/controls/Output K1 both

```

Рисунок 2.6 – Файл конфигурации для подключения к топикам

### 3. Практическая работа №11

Согласно варианту были составлены скрипты с различными тревогами.

В первом случае при превышении концентрации CO<sub>2</sub> определённого значения отправляется уведомление с текущим уровнем CO<sub>2</sub>.

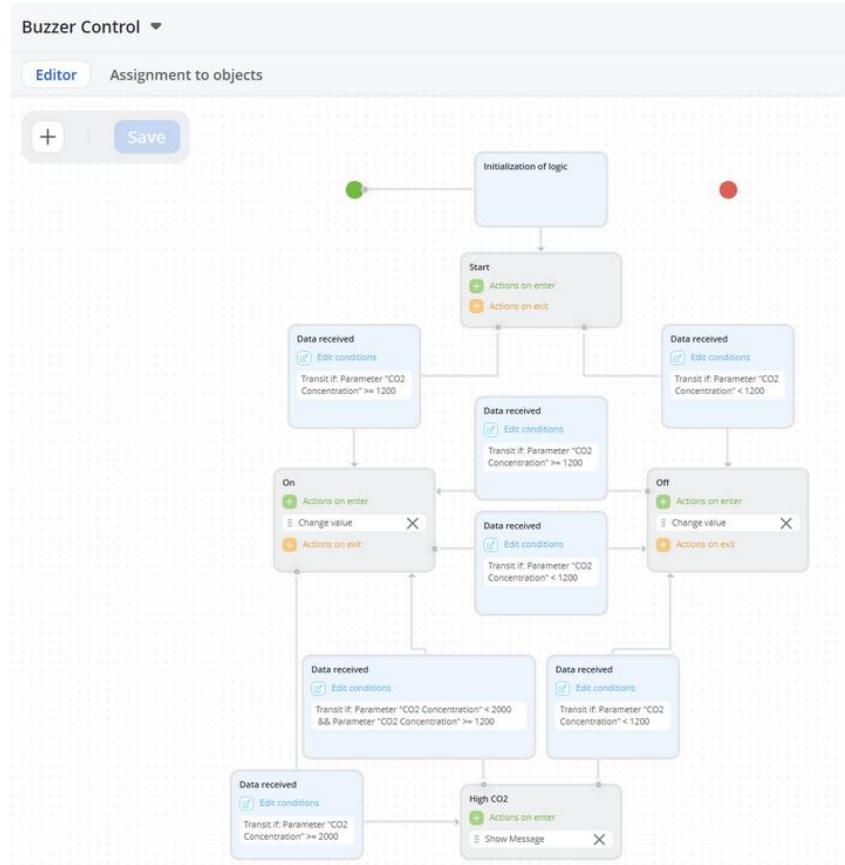


Рисунок 3.1 – Правило автомата первого задания

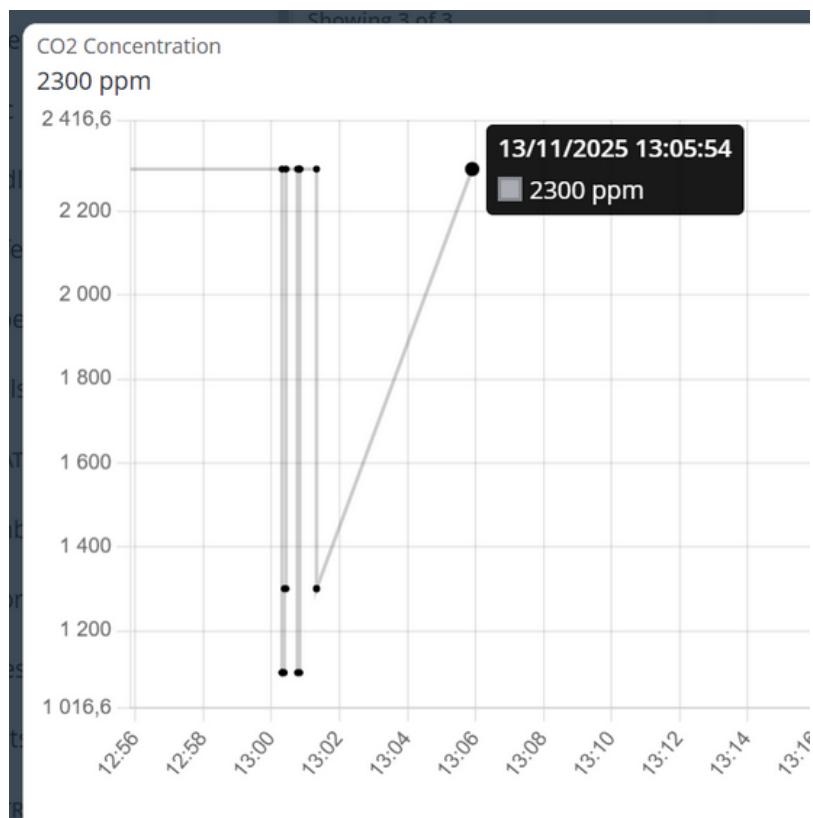


Рисунок 3.2 – Зафиксированные значения датчика концентрации СО<sub>2</sub>

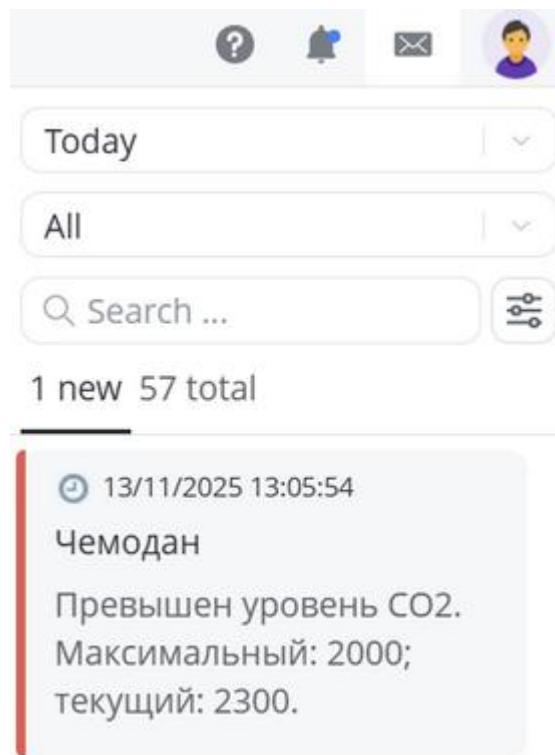


Рисунок 3.3 – Результат тестирования автомата первого задания

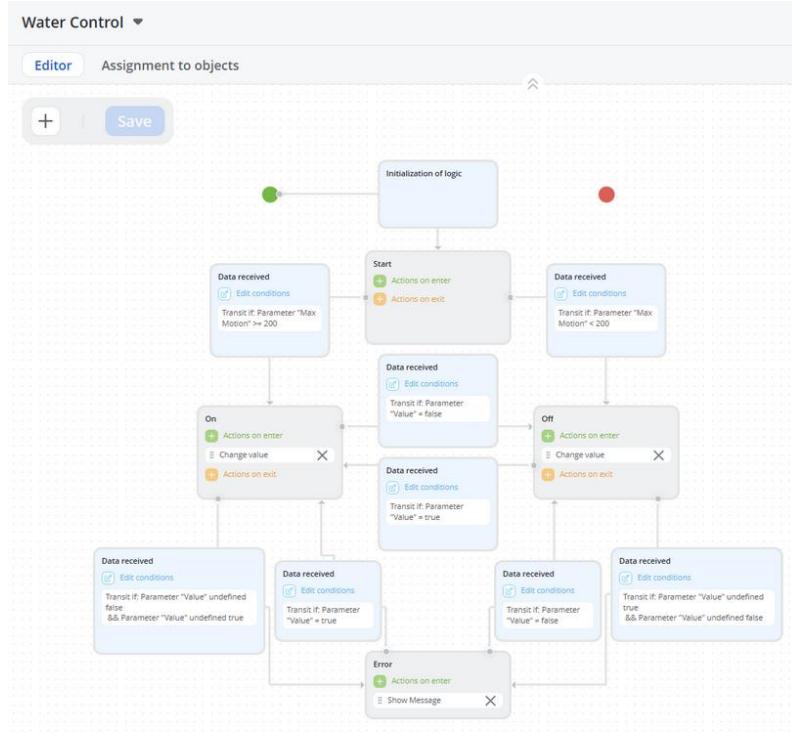


Рисунок 3.4 – Правило автомата для второго задания

Во втором задании при передаче данных, в которых нет motion, пользователю передаётся сообщение об ошибке.

```

Handlers << + 
Search ... 
No filters >
Showing 1 of 1
Анализатор движение...
JS 1
1 export function process(topic, payload) {
2   const threshold = 200;
3   if (topic != "/devices/wb-msw-v3_64/controls/Max Motion")
4   {
5     return { topic: topic }
6   }
7   return { topic: topic, value: payload > threshold}
8 }

```

Рисунок 3.5 – Обработчик для второго правила

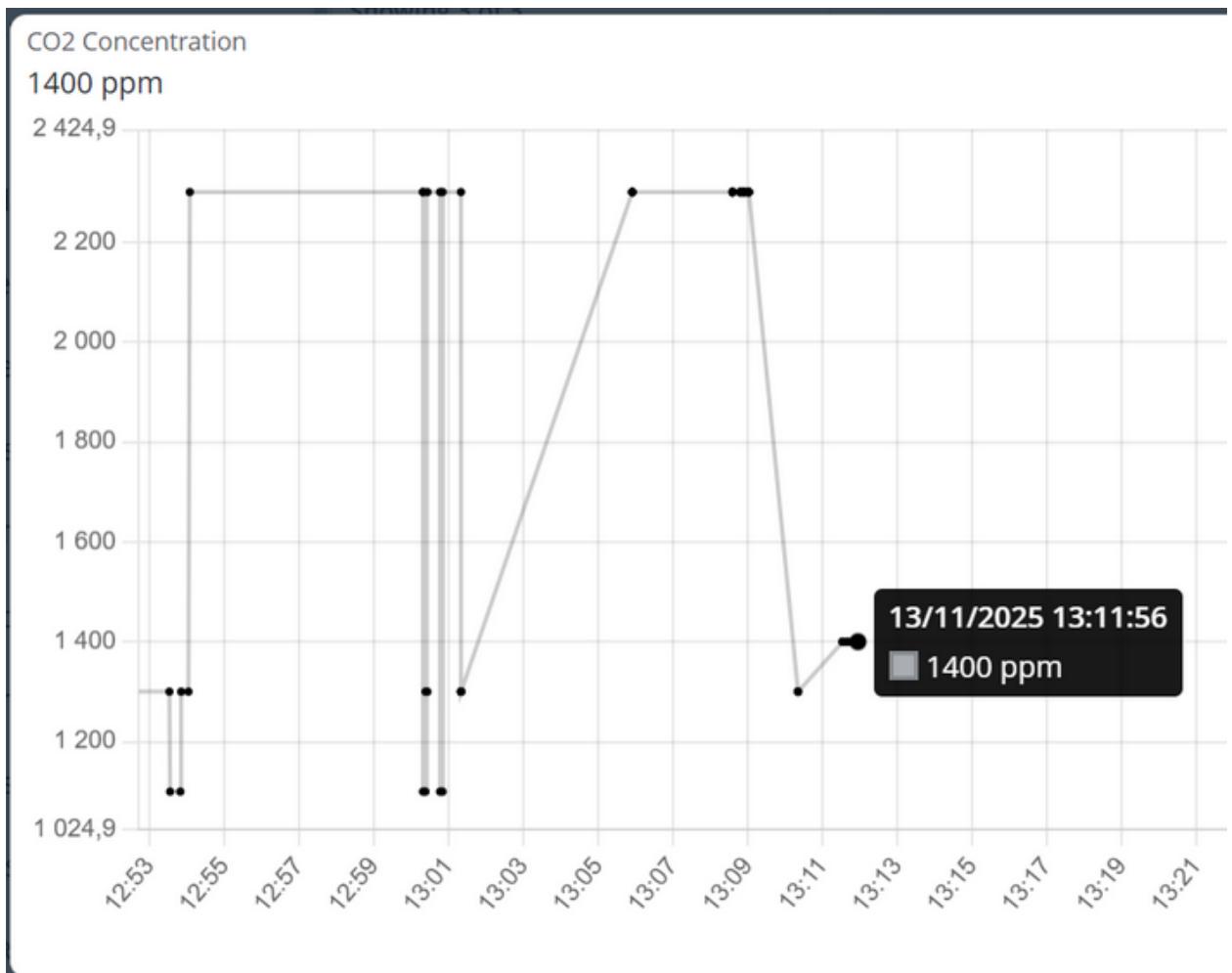


Рисунок 3.6 – Переданные данные в обработчик

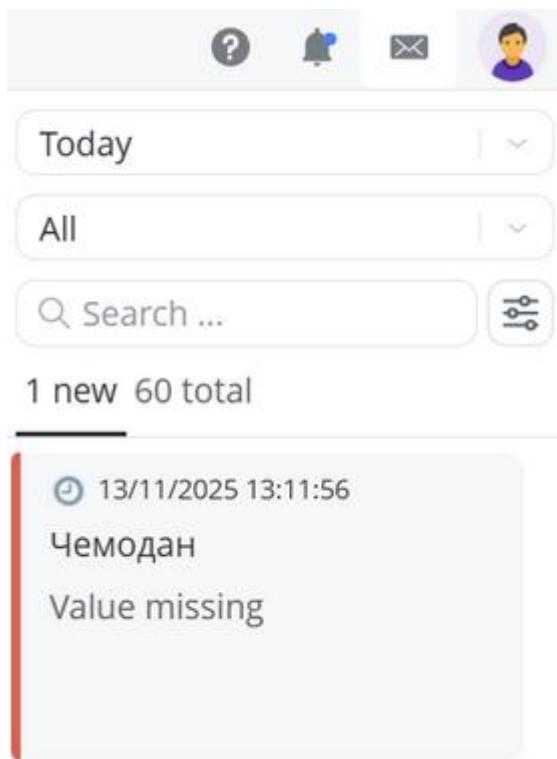


Рисунок 3.7 – Результат тестирования автомата второго задания

#### 4. Практическая работа №12

Перед началом выполнения практической работы произошло настрой SMTP сервера.

The screenshot shows the 'Project settings' page for a project named 'Project of mihakom050@gmail.com'. The 'Integrations' tab is selected. A section titled 'Сообщения' (Messages) is highlighted. The configuration details for the 'Сообщения' channel are as follows:

Message channel name *	Сообщения
Description	Describe the integration
Host *	smtp.gmail.com
Port *	587
Username *	mihakom050@gmail.com
Password *	.....
Sender	IoT Message
Connection security	STARTTLS

At the bottom are 'Save' and 'Remove' buttons.

Рисунок 4.1 – Настройка SMTP сервера  
Далее было настроено отправляемое сообщение.

The 'Send e-mail' configuration dialog is shown. The 'SMTP gateway' is set to 'Сообщения'. The 'From' field contains 'mihakom050@gmail.com'. The 'To' field also contains 'mihakom050@gmail.com'. The 'Subject' field is set to 'High CO2'. The 'Mail body (text)' field contains the message: 'Превышен уровень CO2. Максимальный: 2000; текущий: {  
{объект.state["co2"]}}.' Below the body is an 'Add parameter' button.

Рисунок 4.2 – Настройка отправки ошибочного сообщения с концентрацией CO2

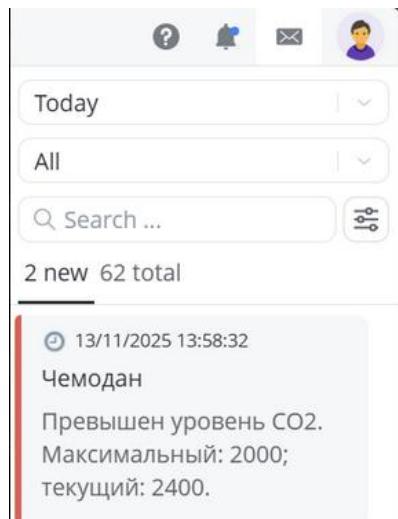


Рисунок 4.3 – Пример итогового ошибочного сообщения с концентрацией CO<sub>2</sub>

И в конечном итоге был проведён тест корректности работы топиков.



Рисунок 4.4 – Получено письмо с ошибкой о концентрации CO<sub>2</sub>  
Далее было настроено следующее сообщение в соответствии с заданием.

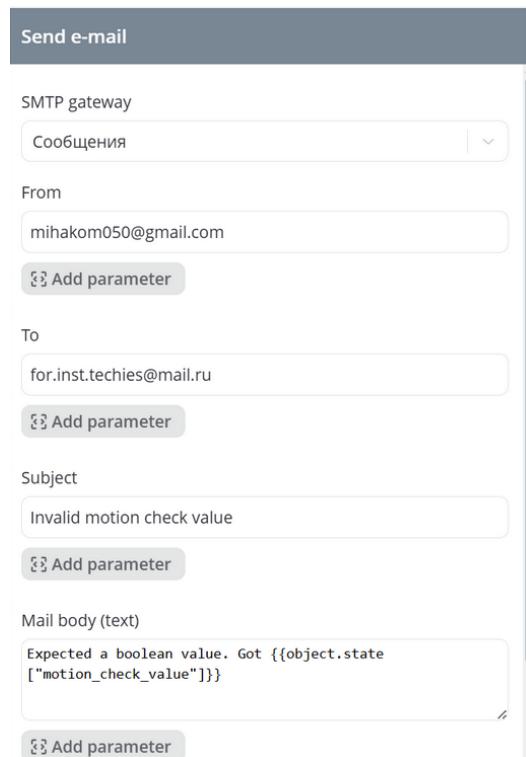


Рисунок 4.5 – Настройка отправки ошибочного сообщения с наличием motion

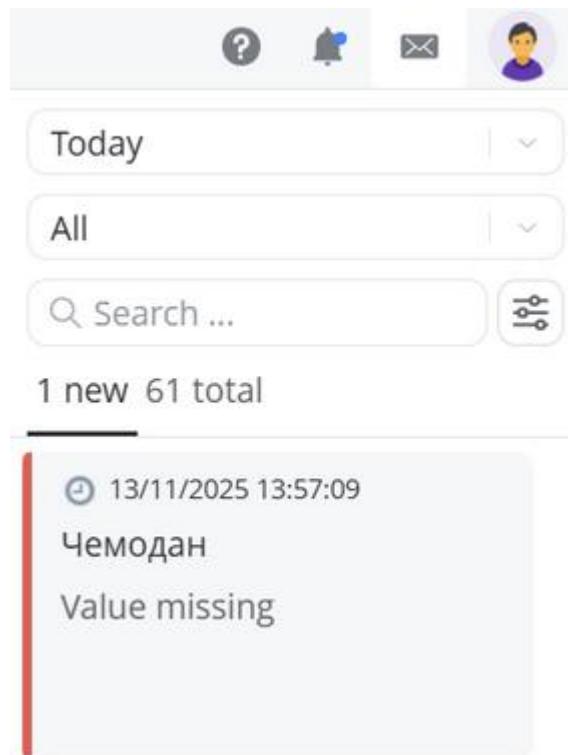


Рисунок 4.6 – Пример итогового ошибочного сообщения с motion

При этом была проведена проверка корректности работы сообщений отправкой сообщения на почту.

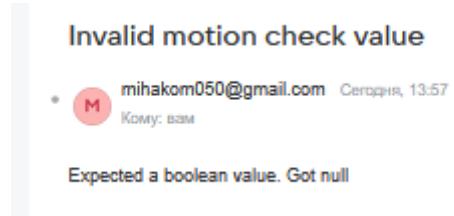


Рисунок 4.7 – Получено письмо с ошибкой о motion

## **ВЫВОД**

В ходе данных практических работ было проведено ознакомление с функциональными возможностями сайта dev.rightech.io и проведена работа с динамически меняющимися данными. Полученные знания были закреплены путём выполнения практических заданий.