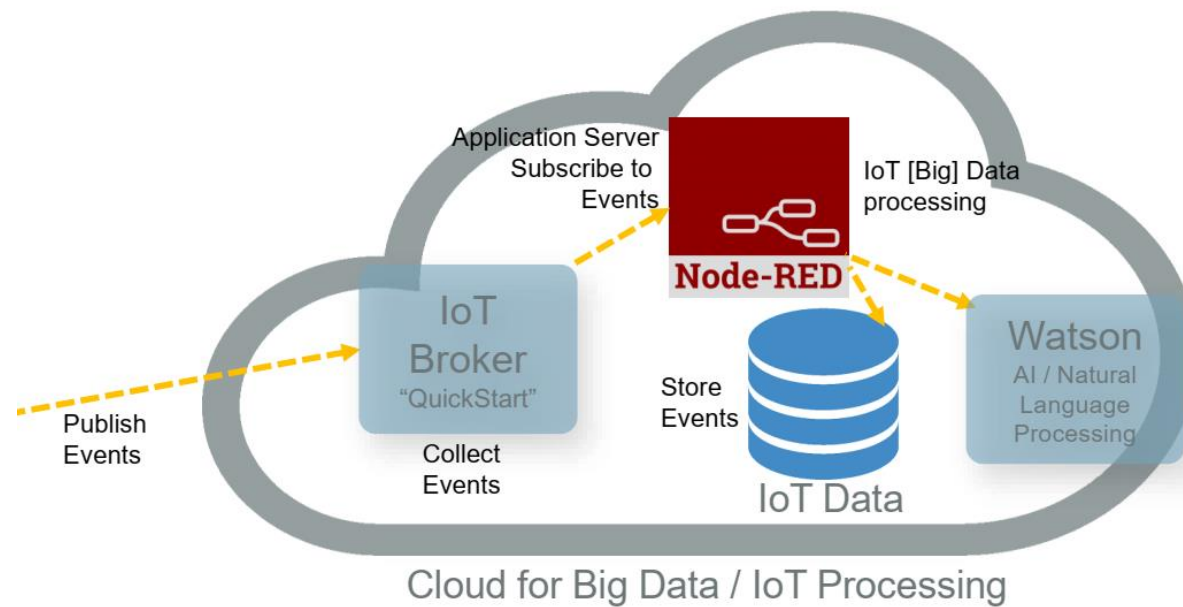


Практическая работа №2 Проектирование и моделирование системы IoT

В этой практической работе вы создадите приложение Node-RED для сбора, хранения и отображения данных виртуальных датчиков



1. Создайте флоу в Node-Red, отправляющий данные из Sense HAT Simulator Device Simulator в Node-Red.

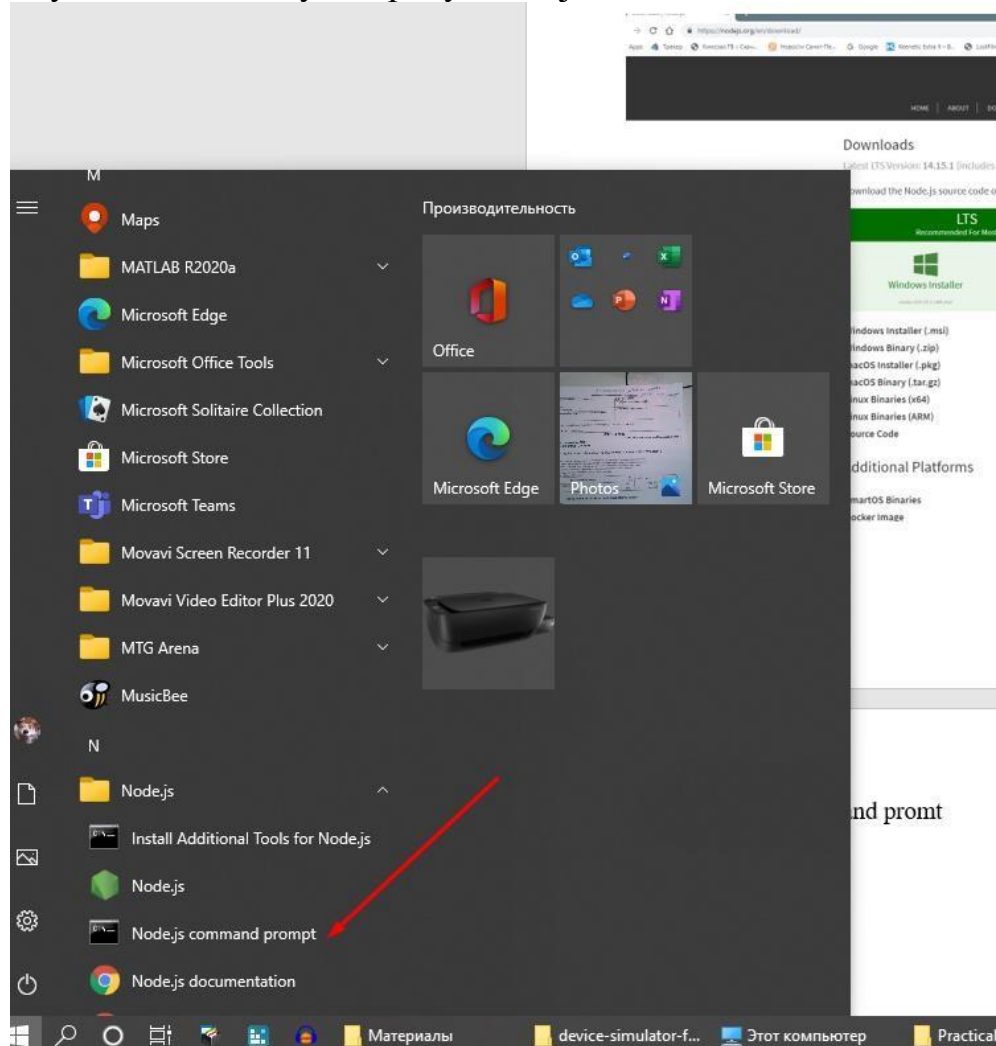
Сначала вам нужно установить на свой компьютер Node.js <https://nodejs.org/en/download/> Выберите правильную версию для вашей операционной системы.

The screenshot shows the Node.js download page. The browser address bar displays <https://nodejs.org/en/download/>. The page features a dark header with the Node.js logo and navigation links: HOME, ABOUT, DOWNLOADS, DOCS, GET INVOLVED, SECURITY, CERTIFICATION, and NEWS. Below the header, the 'Downloads' section highlights the 'Latest LTS Version: 14.15.1 (includes npm 6.14.8)'. It encourages users to download the source code or a pre-built installer. Two main columns are shown: 'LTS Recommended For Most Users' and 'Current Latest Features'. Under 'LTS', there are links for 'Windows Installer' (node-v14.15.1-x64.msi), 'macOS Installer' (node-v14.15.1.pkg), and 'Source Code' (node-v14.15.1.tar.gz). Under 'Current', there are links for 'Windows Installer (.msi)', 'Windows Binary (.zip)', 'macOS Installer (.pkg)', 'macOS Binary (.tar.gz)', 'Linux Binaries (x64)', 'Linux Binaries (ARM)', and 'Source Code'. A table below these links lists the available binaries for each platform. For Windows, there are 32-bit and 64-bit versions. For macOS, there are 64-bit versions. For Linux, there are 64-bit versions and ARMv7/ARMv8 versions. The 'Source Code' link points to node-v14.15.1.tar.gz. The 'Additional Platforms' section lists 'SmartOS Binaries' (64-bit) and 'Docker Image' (Official Node.js Docker Image).

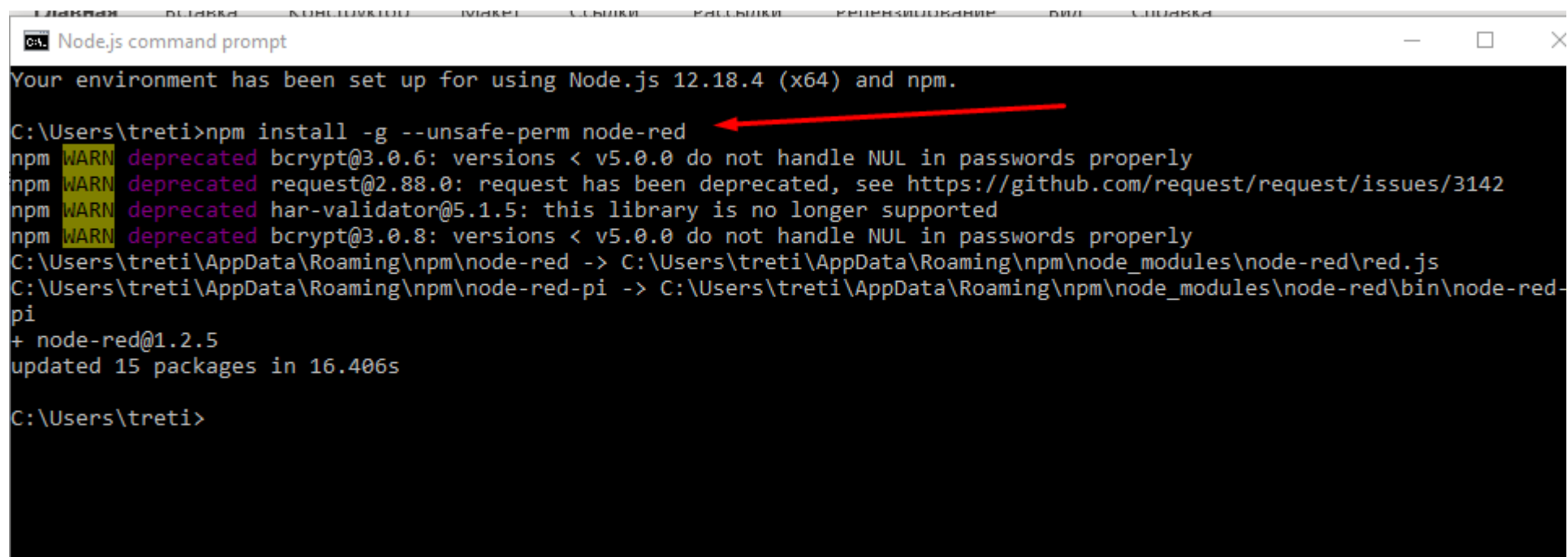
Platform	Architecture	Download Link
Windows	32-bit	node-v14.15.1-x64.msi
	64-bit	node-v14.15.1-x64.msi
macOS	64-bit	node-v14.15.1.pkg
	64-bit	node-v14.15.1.pkg
Linux	64-bit	node-v14.15.1.tar.gz
	ARMv7	node-v14.15.1.tar.gz
Linux	ARMv8	node-v14.15.1.tar.gz
	node-v14.15.1.tar.gz	

Platform	Architecture	Download Link
SmartOS	64-bit	node-v14.15.1.tar.gz
	64-bit	node-v14.15.1.tar.gz
Docker	64-bit	node-v14.15.1.tar.gz
	64-bit	node-v14.15.1.tar.gz

Запустите командную строку Node.js



В открывшемся окне наберите - `npm install -g --unsafe-perm node-red`



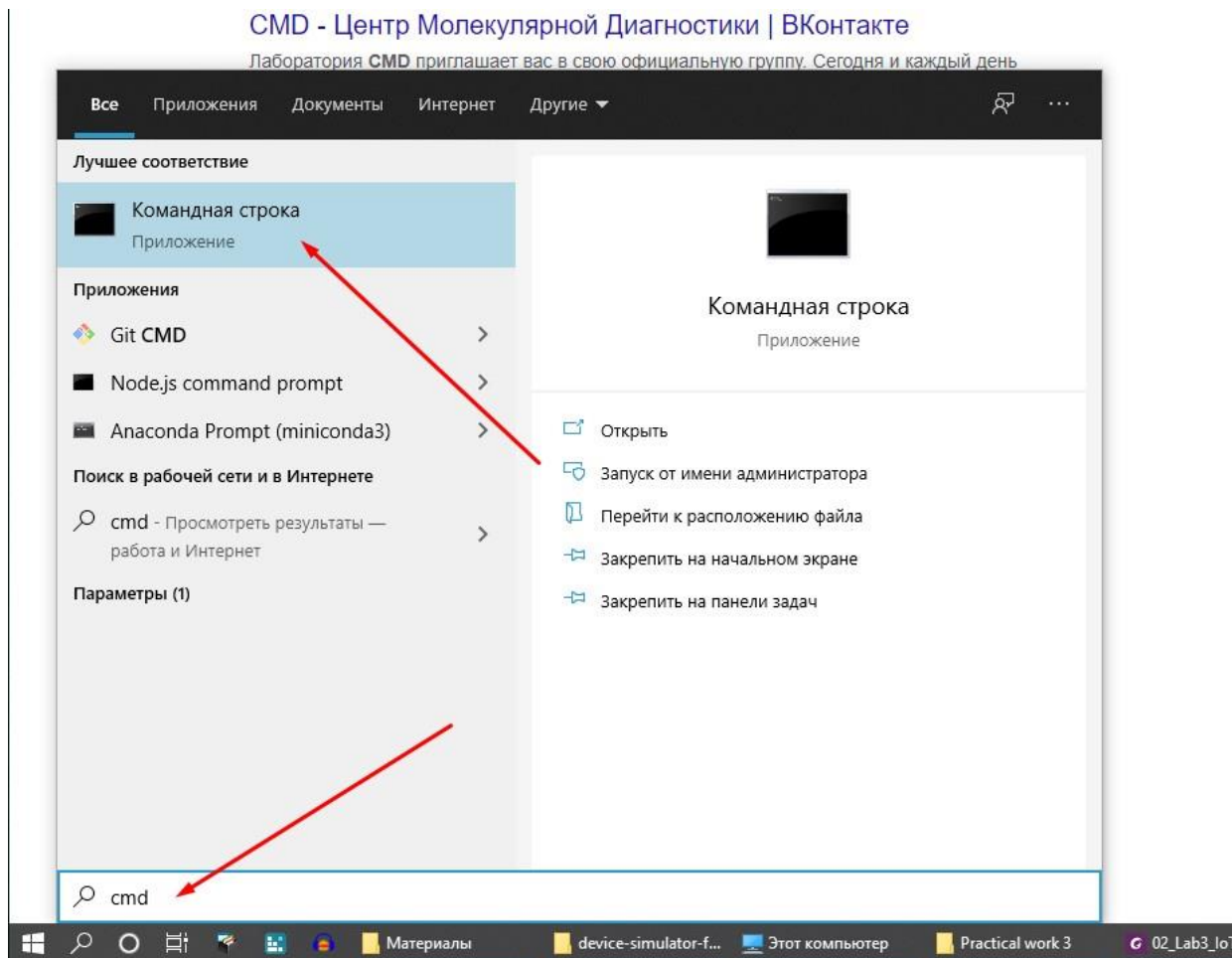
```
Node.js command prompt
Your environment has been set up for using Node.js 12.18.4 (x64) and npm.

C:\Users\treti>npm install -g --unsafe-perm node-red
npm WARN deprecated bcrypt@3.0.6: versions < v5.0.0 do not handle NUL in passwords properly
npm WARN deprecated request@2.88.0: request has been deprecated, see https://github.com/request/request/issues/3142
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
npm WARN deprecated bcrypt@3.0.8: versions < v5.0.0 do not handle NUL in passwords properly
C:\Users\treti\AppData\Roaming\npm\node-red -> C:\Users\treti\AppData\Roaming\npm\node_modules\node-red\red.js
C:\Users\treti\AppData\Roaming\npm\node-red-pi -> C:\Users\treti\AppData\Roaming\npm\node_modules\node-red\bin\node-red-pi
+ node-red@1.2.5
updated 15 packages in 16.406s

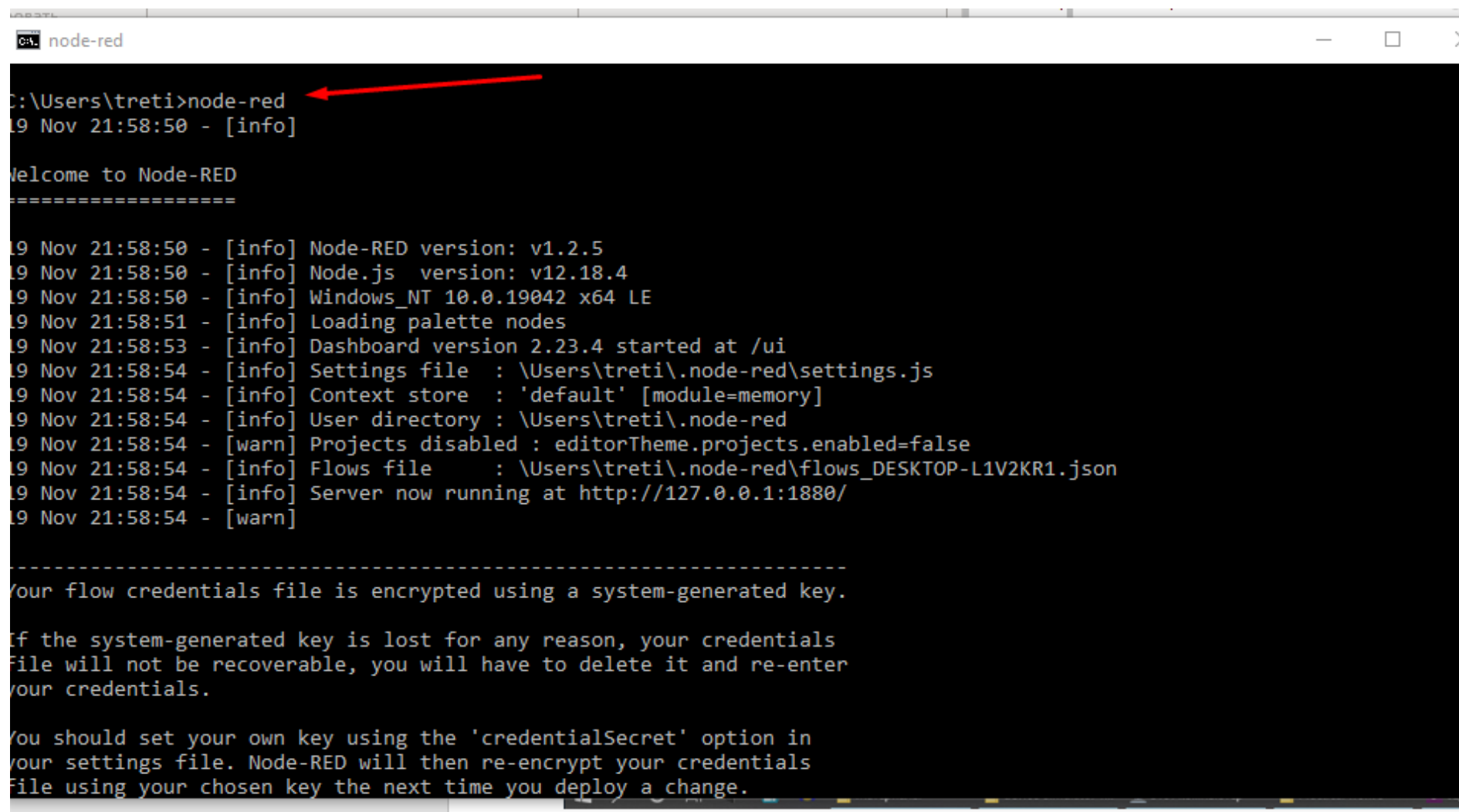
C:\Users\treti>
```

Инсталляция для других операционных систем: <https://nodered.org/docs/getting-started/local>

Введите в поиске Windows cmd и откройте командную строку Windows по умолчанию.



Введите Node-red



```
C:\Users\treti>node-red
19 Nov 21:58:50 - [info]

Welcome to Node-RED
=====

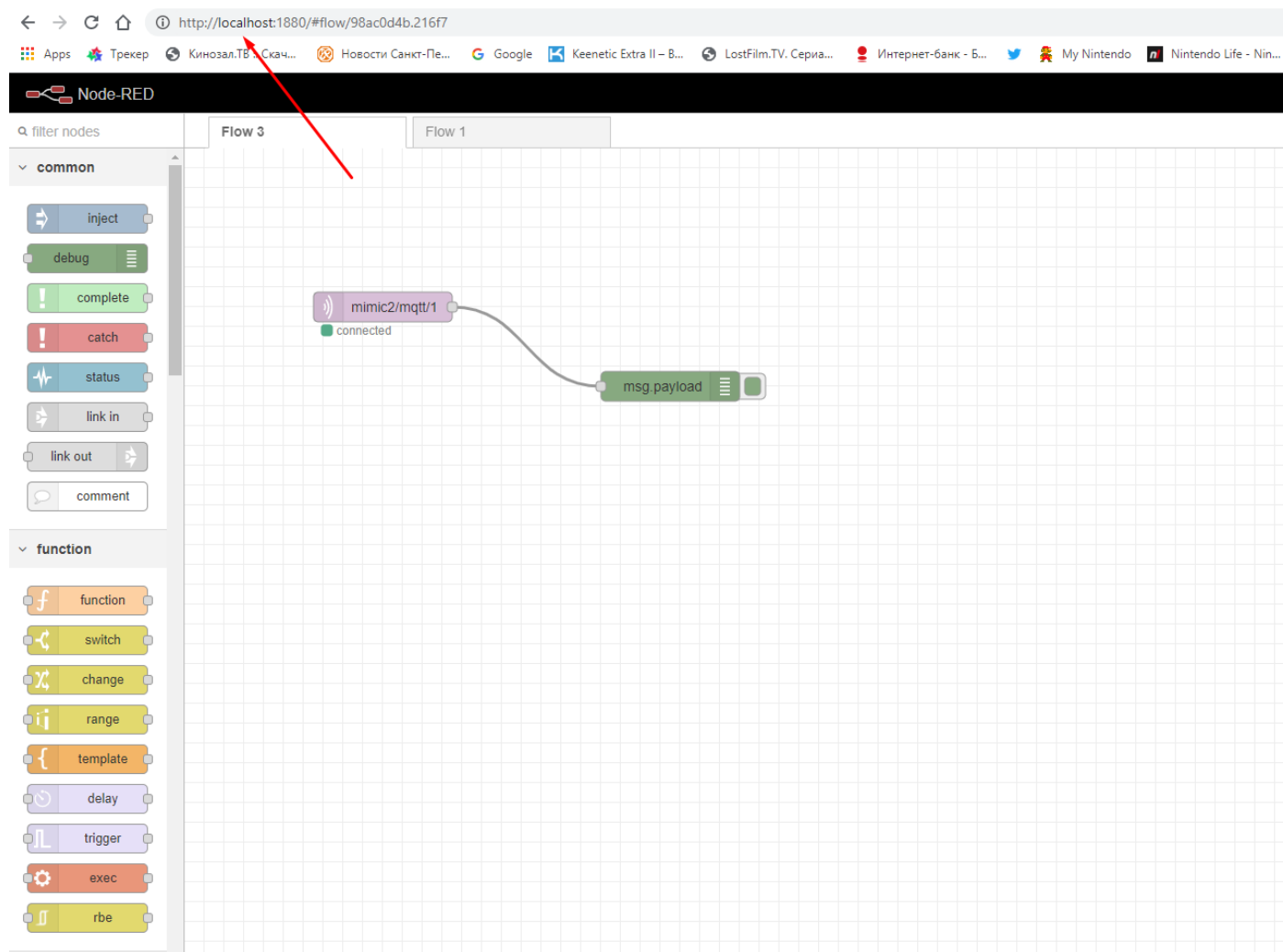
19 Nov 21:58:50 - [info] Node-RED version: v1.2.5
19 Nov 21:58:50 - [info] Node.js version: v12.18.4
19 Nov 21:58:50 - [info] Windows_NT 10.0.19042 x64 LE
19 Nov 21:58:51 - [info] Loading palette nodes
19 Nov 21:58:53 - [info] Dashboard version 2.23.4 started at /ui
19 Nov 21:58:54 - [info] Settings file : \Users\treti\.node-red\settings.js
19 Nov 21:58:54 - [info] Context store : 'default' [module=memory]
19 Nov 21:58:54 - [info] User directory : \Users\treti\.node-red
19 Nov 21:58:54 - [warn] Projects disabled : editorTheme.projects.enabled=false
19 Nov 21:58:54 - [info] Flows file : \Users\treti\.node-red\flows_DESKTOP-L1V2KR1.json
19 Nov 21:58:54 - [info] Server now running at http://127.0.0.1:1880/
19 Nov 21:58:54 - [warn]

-----
Your flow credentials file is encrypted using a system-generated key.

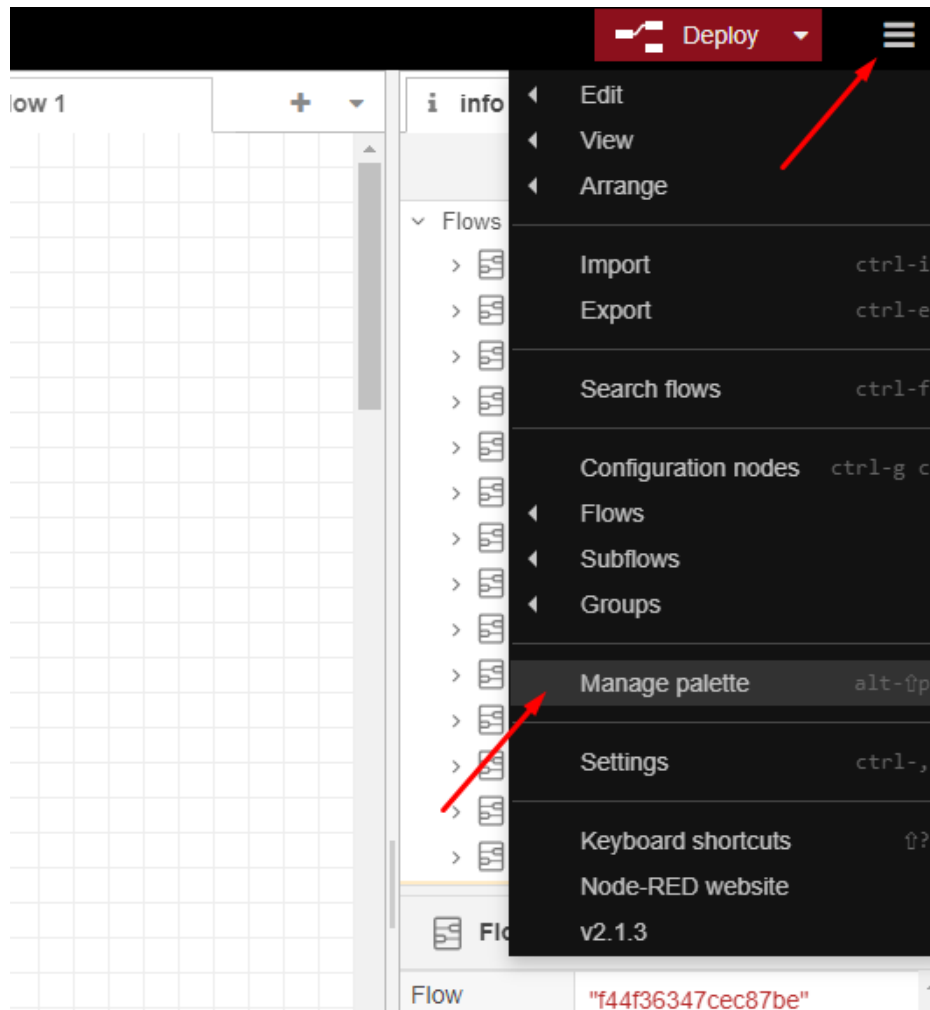
If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
```

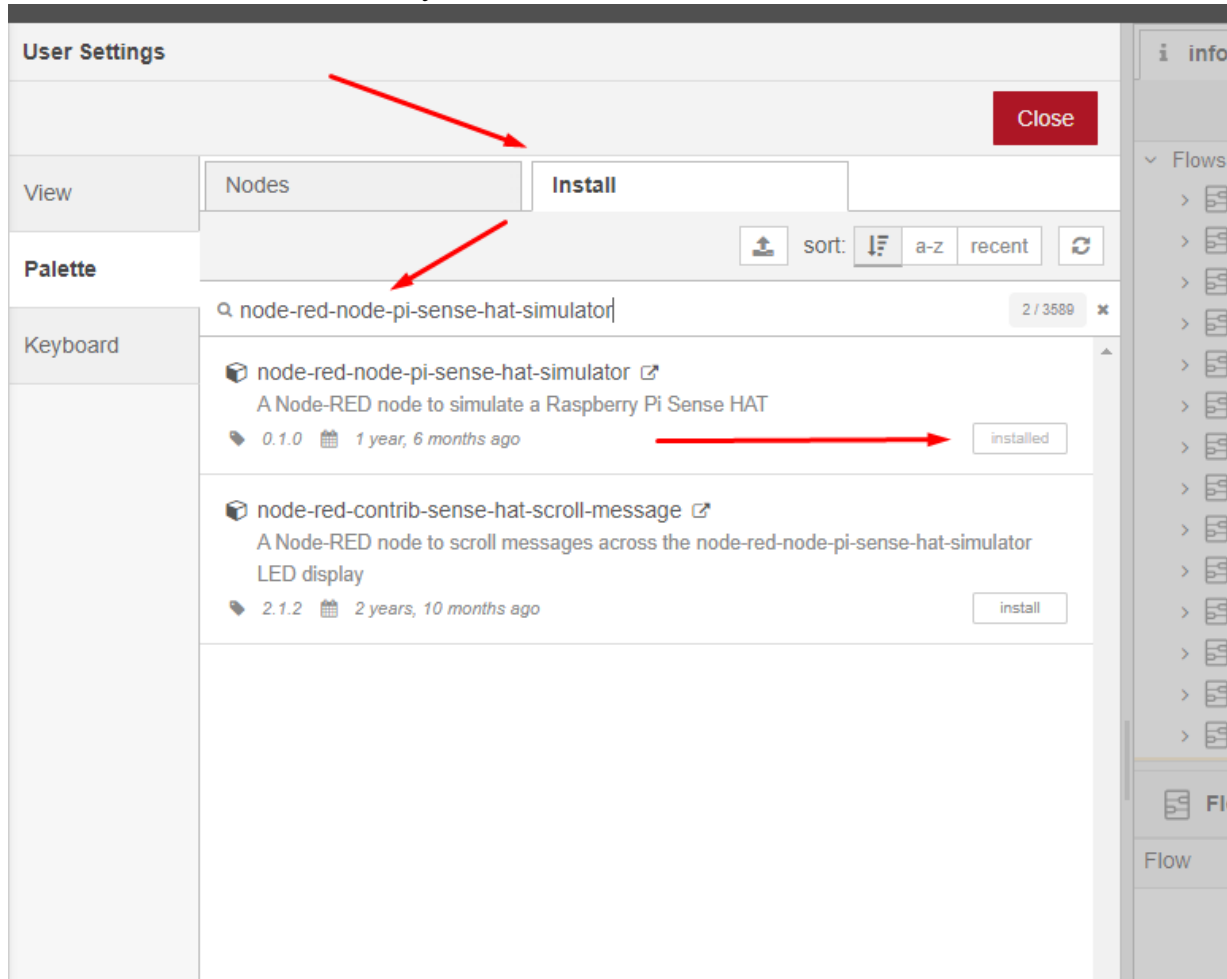
Откройте интернет-браузер и введите <http://localhost:1880>



Теперь давайте добавим ноду Sense HAT Simulator из Palette. Сначала нажмите на знак меню «Бургер» и выберите «Manage palette».



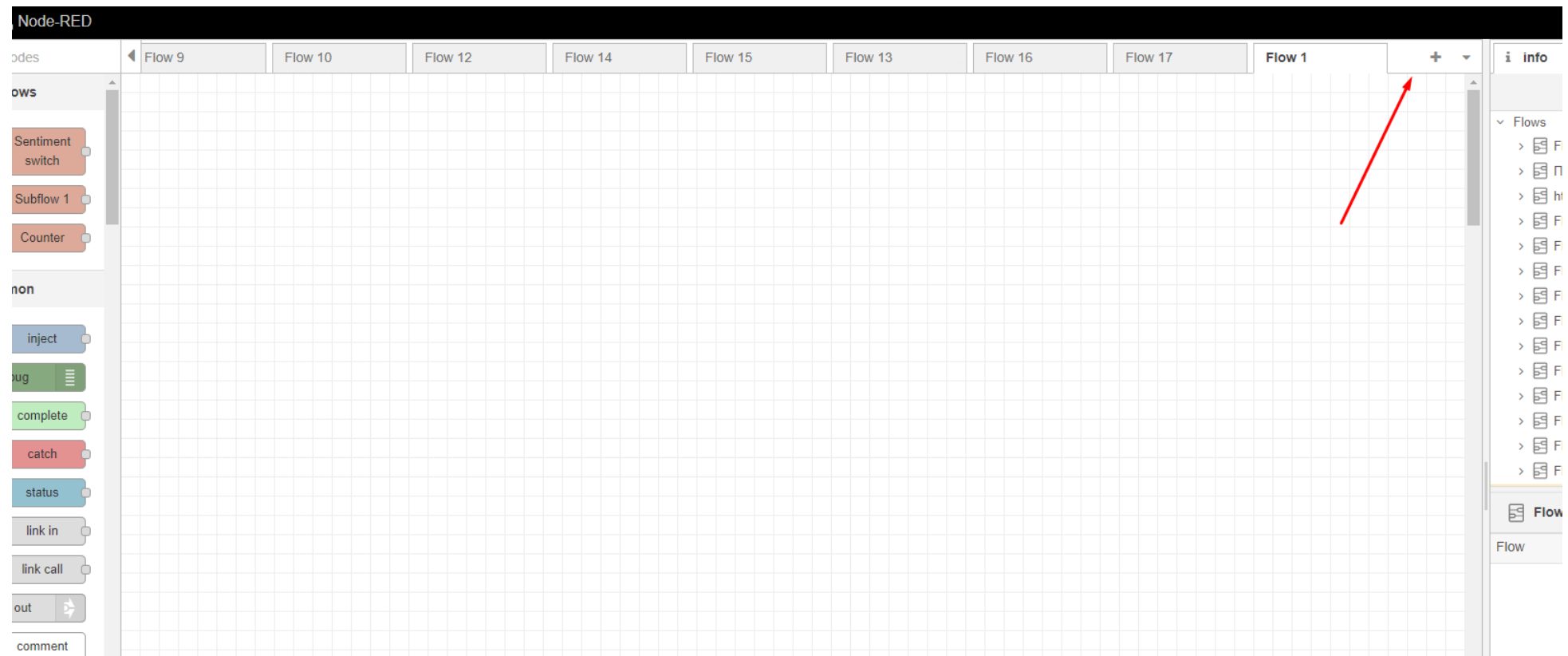
Затем нажмите на панель установки и введите в окне поиска «node-red-node-pi-sense-hat-simulator» и установите его.



Pi-sense-hat-simulator — это нода для моделирования Raspberry Pi Sense HAT.

Используя данный симулятор, мы сможем создавать приложения, которые взаимодействуют с виртуальной средой Sense HAT без реального оборудования — независимо от того, работаете ли вы на Raspberry Pi, ноутбуке или где-либо еще.

Теперь давайте создадим новый флоу, нажав кнопку +:



И в меню нод на левой панели выберите нод Sense HAT Sim с выходным квадратом справа. Перетащите его в рабочую область.



Теперь нажмите Sense HAT Sim, чтобы выделить его, и нажмите кнопку HELP на правой панели.

The screenshot displays the Node-RED web interface. At the top, there are tabs for 'Flow 9', 'Flow 10', and 'Flow 12'. A red arrow points from the 'node-red-node-pi-sense-hat-simulator' entry in the help sidebar to the 'Sense HAT Sim' node in the workspace. Another red arrow points from the 'Open Simulator' button in the help panel to the right side of the 'Sense HAT Sim' node. The help panel on the right contains the following text:

Sense HAT Sim

Raspberry Pi Sense HAT Simulator input node.

This node simulates readings from the various sensors on the Sense HAT, grouped into three sets; motion events, environment events and joystick events.

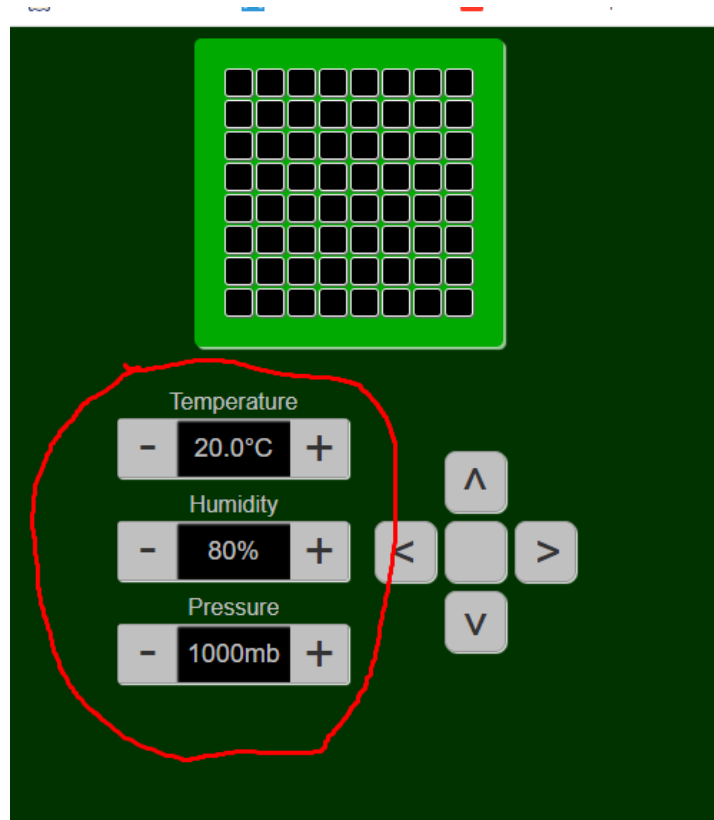
Once deployed, click this button to open the simulator:

[Open Simulator ↗](#)

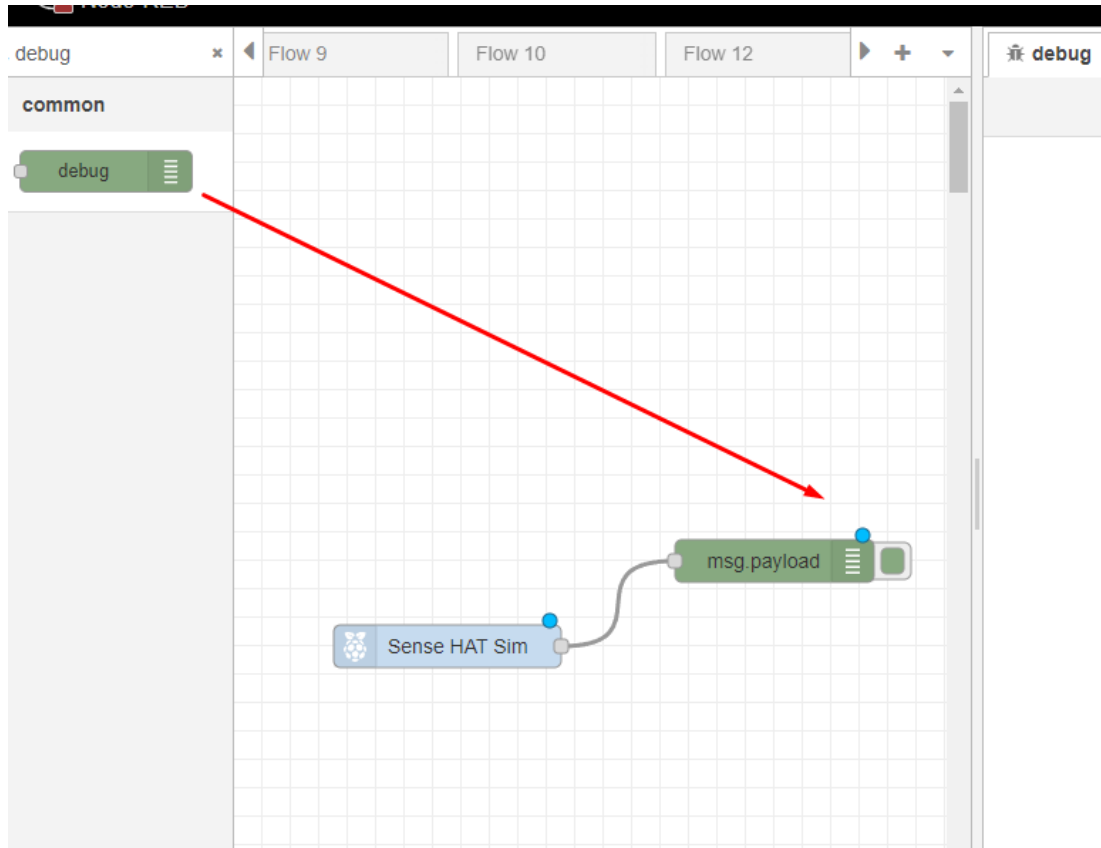
Motion events - *not currently supported by the simulator*

Motion events include readings from the accelerometer, gyroscope and magnetometer, as well as the current compass heading. They are

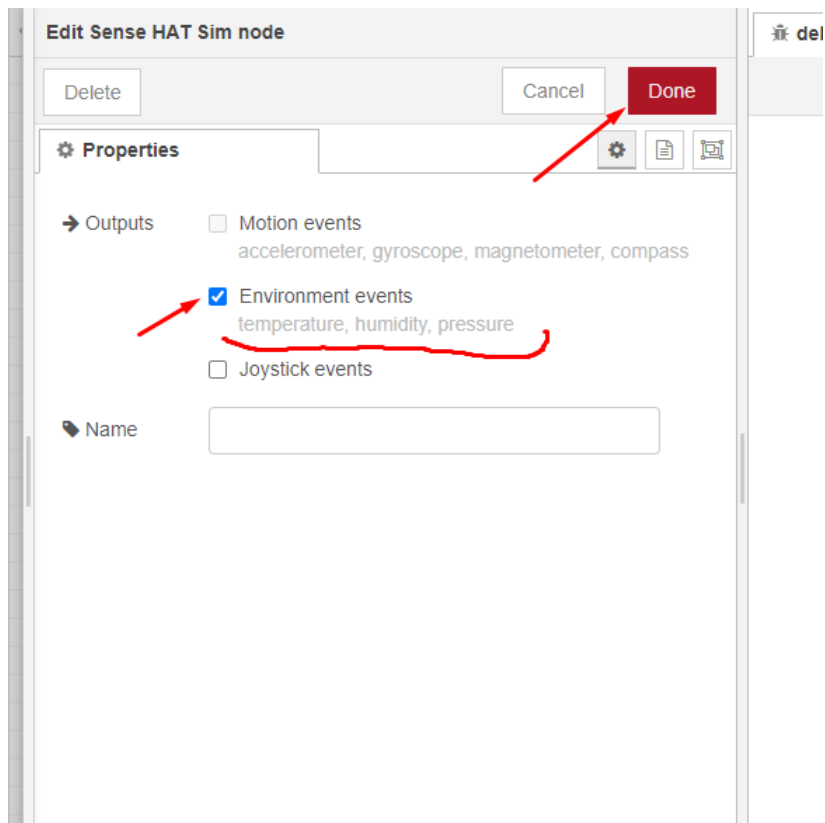
Здесь вы можете увидеть некоторую информацию о ноде симулятора pi NAT и открыть симулятор для имитации данных. Нажмите кнопку «Открыть симулятор», чтобы открыть его в новом окне браузера. Вы можете увидеть там 3 параметра: Температура, Влажность и Давление. Нажимая на кнопки + или – вы можете изменить значения этих параметров.



Теперь добавьте debug node и соедините их вместе.



Дважды щелкните по ноде Sense HAT Sim, чтобы открыть её свойства. Отключите события джойстика, убедитесь, что события среды включены, и нажмите «Done».

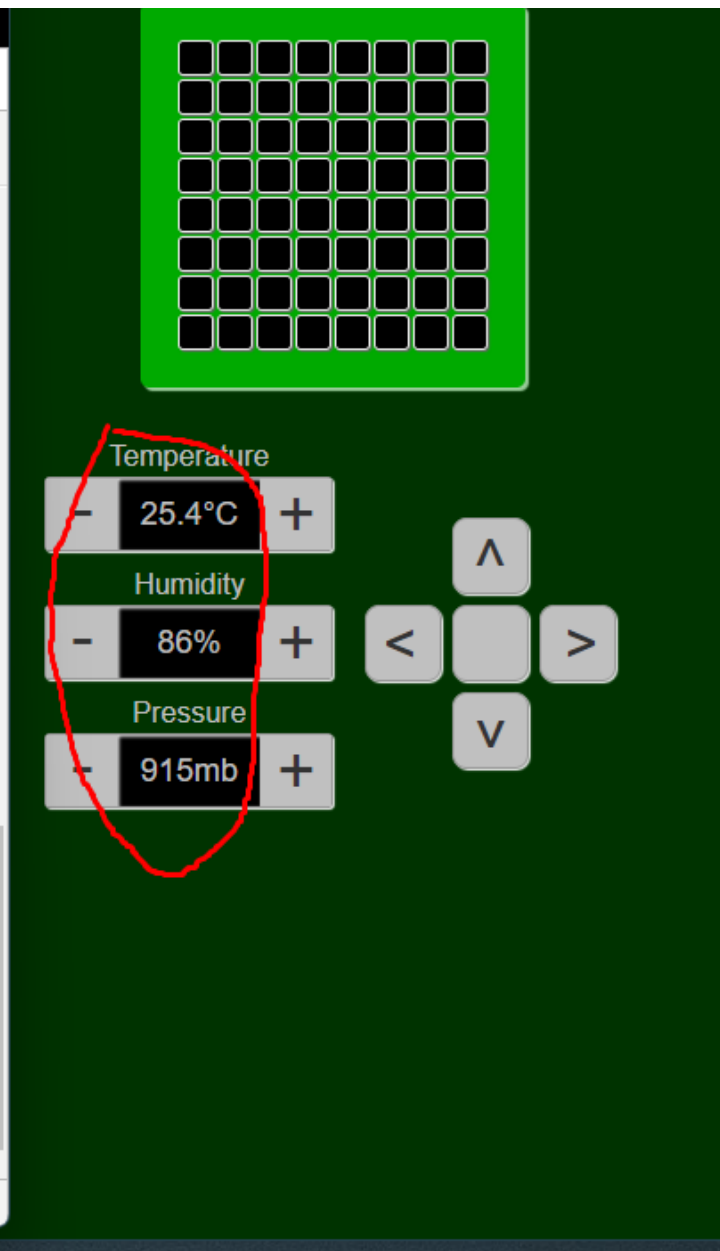
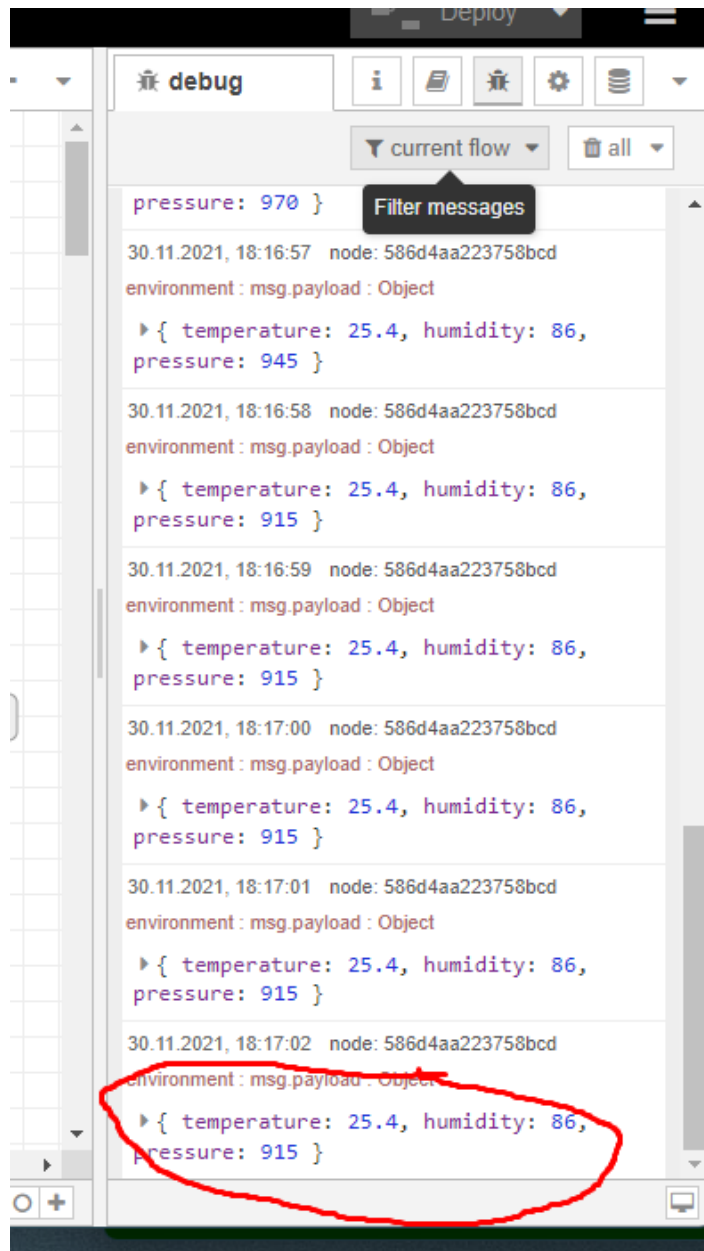


Теперь запустите свой флоу и откройте окно отладки на правой панели. Там вы можете увидеть сообщение с температурой, влажностью и давлением в формате json.

Node-RED interface showing a flow with a **Sense HAT Sim** node connected to a **msg.payload** node. The right sidebar displays the debug console with two log entries:

```
30.11.2021, 18:15:23 node: 586d4aa223758bcd  
environment : msg.payload : Object  
  { temperature: 20, humidity: 80,  
    pressure: 1000 }  
30.11.2021, 18:15:24 node: 586d4aa223758bcd  
environment : msg.payload : Object  
  { temperature: 20, humidity: 80,  
    pressure: 1000 }
```

Теперь попробуем изменить эти параметры и проверим, как меняется сообщение.



2. Создайте флоу для определения граничных значений симуляции событий внешней среды

Теперь выберем значения температуры и составим флоу для определения критических значений температуры. Сначала мы добавляем в наш флоу function node и пишем туда функцию `return{payload:msg.payload.temperature}`:

The screenshot displays the Node-RED web interface. On the left, a flow canvas shows a sequence of nodes: a 'Sense HAT Sim' node (labeled 'connected'), followed by a 'function' node, and then a 'msg.payload' node. A red arrow points from the 'function' node in the canvas to the 'Edit function node' panel on the right. The 'Edit function node' panel has a 'Name' field containing 'Temperature'. Below this, the 'On Message' tab is selected, showing a code editor with the following JavaScript code:

```
1 return{payload:msg.payload.temperature};
2
```

Red arrows also point to the 'Done' button in the top right of the panel and to the first line of code in the editor.

The screenshot shows the Node-RED web interface. At the top, there's a 'Deploy' button with a red arrow pointing to it. Below the top bar, there are tabs for 'Flow 8', 'Flow 9', and 'Flow 10'. The main workspace contains a flow with three nodes: a 'Sense HAT Sim' node (blue) with a 'connected' status, a 'Temperature' function node (orange), and a 'msg.payload' output node (green dashed box). The debug console on the right is open, showing a list of messages. The first message is '25.4'. The second message is '30.11.2021, 18:29:21 node: 586d4aa223758bcd msg.payload : number 25.4'. The third message is '30.11.2021, 18:29:22 node: 586d4aa223758bcd msg.payload : number 25.4'. The fourth message is '30.11.2021, 18:29:23 node: 586d4aa223758bcd msg.payload : number 25.4'. The fifth message is '30.11.2021, 18:29:24 node: 586d4aa223758bcd msg.payload : number 25.4'. The sixth message is '30.11.2021, 18:29:26 node: 586d4aa223758bcd msg.payload : number 25.4'. The seventh message is '30.11.2021, 18:29:27 node: 586d4aa223758bcd msg.payload : number 25.4'. The value '25.4' in the sixth message is circled in red.

Теперь вы можете видеть в окне отладки только значения температуры.

Теперь мы добавим Switch node и зададим температурный порог. Например: ≤ 40 Температура в безопасных пределах; > 40 Критическая температура:

Node-RED

Flow 8 Flow 9 Flow 10

subflows

- Sentiment switch

function

- switch

dashboard

- switch

Sense HAT Sim

connected

Temperature

switch

Edit switch node

Delete Cancel Done

Properties

Name Temperature Threshold

Property msg. payload

3

2

6

1

4

5

+

add

checking all rules

```
graph LR; SH[Sense HAT Sim] --> T[Temperature]; T --> S[switch]; S --> S;
```

И добавьте несколько нод Templates для обмена сообщениями:

The screenshot displays the Node-RED web interface. On the left, the 'function' and 'dashboard' palettes are visible. The main workspace shows a flow diagram with the following components:

- Sense HAT Sim** (blue node) connected to **Temperature** (orange node) and **Temperature Threshold** (yellow node).
- Temperature** node connected to a **Safe** template node (orange node with curly braces).
- Temperature Threshold** node connected to a **Critical** template node (orange node with curly braces).
- Both **Safe** and **Critical** nodes connected to a **msg.payload** output node (green node).

On the right, the **Edit template node** panel is open for the **Safe** node. It contains the following fields:

- Name:** Safe
- Property:** msg.payload
- Template:** 1 `{{payload}}` temperature within safe limits
- Syntax Highlight:** mustache
- Format:** Mustache template

Red arrows indicate the workflow: one arrow points from the **template** node in the left palette to the **Safe** node in the flow; another arrow points from the **template** node in the left palette to the **Critical** node; a third arrow points from the **Safe** node in the flow to the **Safe** template node in the edit panel.

The screenshot shows the Node-RED editor interface. On the left, a flow diagram features two nodes labeled 'Safe' and 'Critical' connected to a central 'msg.payload' node. A red arrow originates from the 'Critical' node and points towards the 'Properties' panel on the right. The 'Properties' panel is open, displaying the following configuration:

- Name:** Critical
- Property:** msg.payload
- Template:** `{{payload}}` Warning - temperature critical !
- Syntax Highlight:** mustache
- Format:** Mustache template
- Output as:** Plain text

At the top right of the 'Properties' panel, there are three buttons: 'Delete', 'Cancel', and 'Done'. A red arrow points to the 'Done' button. The 'Template' field contains a red arrow pointing to the `{{payload}}` placeholder.

Запустите свой флоу. Теперь вы можете видеть в панели отладки предупреждение о состоянии температуры. Попробуйте изменить температуру выше порогового значения и проверьте сообщение.

The screenshot displays a Node-RED interface. On the left, the 'debug' console shows a series of messages from a node with ID '586d4aa223758bcd'. Each message has a payload of 'string[36]' and a red text message: '25.4 temperature within safe limits '. A red arrow points from the sixth message in the log to the 'Temperature' control on the dashboard.

On the right, a custom dashboard is visible with a dark green background. At the top is a 5x5 grid of 25 small black squares. Below the grid are three control panels for 'Temperature', 'Humidity', and 'Pressure'. Each panel has a minus button, a text field, and a plus button. The 'Temperature' panel shows '25.4°C', 'Humidity' shows '86%', and 'Pressure' shows '915mb'. To the right of these panels are four directional buttons: up (^), down (v), left (<), and right (>).

The image shows a Node.js debug console on the left and a temperature control UI on the right. The console displays a series of log messages from a device with node ID 586d4aa223758bcd. The messages show temperature readings increasing from 35.4°C to 41.4°C, with warnings for critical temperatures starting at 40.2°C. The UI on the right has a green background and a 5x5 grid of buttons in the top right. It features three sliders for Temperature (41.4°C), Humidity (86%), and Pressure (915mb). A red arrow points from the '41.2 Warning - temperature critical !' log message to the Temperature slider, and another red arrow points from the '41.4 Warning - temperature critical !' log message to the Temperature slider's value display.

debug

current

30.11.2021, 18:46:47 node: 586d4aa223758bcd
msg.payload : string[36]
"35.4 temperature within safe limits "

30.11.2021, 18:46:48 node: 586d4aa223758bcd
msg.payload : string[36]
"36.2 temperature within safe limits "

30.11.2021, 18:46:49 node: 586d4aa223758bcd
msg.payload : string[36]
"37.2 temperature within safe limits "

30.11.2021, 18:46:50 node: 586d4aa223758bcd
msg.payload : string[36]
"38.2 temperature within safe limits "

30.11.2021, 18:46:51 node: 586d4aa223758bcd
msg.payload : string[36]
"39.2 temperature within safe limits "

30.11.2021, 18:46:52 node: 586d4aa223758bcd
msg.payload : string[37]
"40.2 Warning - temperature critical !"

30.11.2021, 18:46:53 node: 586d4aa223758bcd
msg.payload : string[37]
"41.2 Warning - temperature critical !"

30.11.2021, 18:46:54 node: 586d4aa223758bcd
msg.payload : string[37]
"41.4 Warning - temperature critical !"

Temperature
- 41.4°C +

Humidity
- 86% +

Pressure
- 915mb +

Сделайте то же самое для параметров влажности и давления.

3. Создайте визуализацию для событий внешней среды

Теперь давайте создадим некоторую визуализацию для нашего приложения.

Во-первых, давайте добавим линейную диаграмму

The screenshot shows the Node-RED web interface. On the left, a dashboard titled 'dashboard' contains a 'chart' node. A red arrow points from this node to the 'Edit chart node' panel on the right. The panel has tabs for 'Properties', 'Data', and 'Visuals'. The 'Properties' tab is active, showing various settings for the chart. A red arrow points to the 'Group' dropdown menu, which is currently set to '[Home] HDU'. Another red arrow points to the edit icon (pencil) next to the 'Group' dropdown. The 'Type' is set to 'Line chart'. The 'X-axis' is set to 'last 1 hours' or '1000 points'. The 'X-axis Label' is set to 'HH:mm:ss'. The 'Y-axis' has 'min' and 'max' input fields. The 'Legend' is set to 'None' and 'Interpolate' is set to 'linear'. There are also 'Series Colours' and a 'Blank label' field.

Создайте новую группу, нажав на значок карандаша.

Edit chart node > **Edit dashboard group node**

Delete Cancel Update

Properties

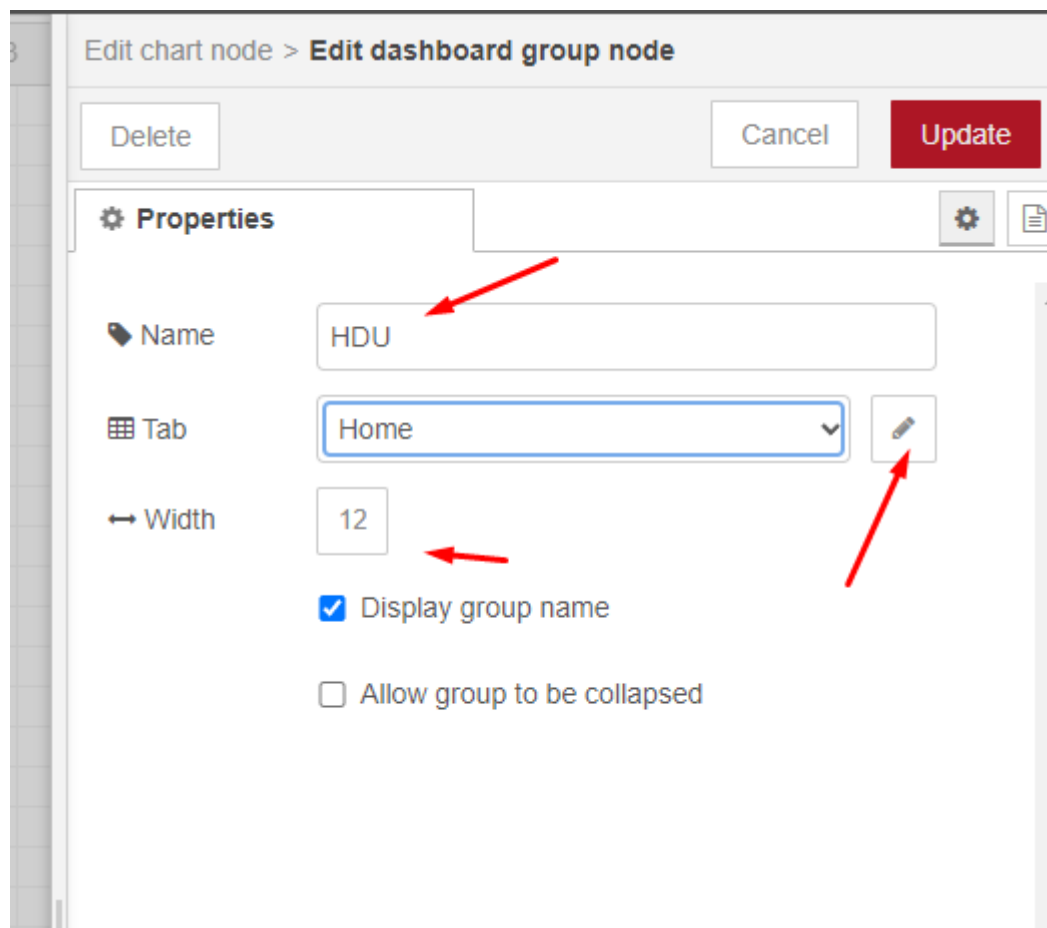
Name HDU

Tab Home

Width 12

☒ Display group name

☐ Allow group to be collapsed



Нажмите на знак карандаша на следующем экране, если вы хотите создать новую вкладку. Придумайте название.

Edit chart node > Edit dashboard group node > **Edit dashboard tab node**

Delete Cancel Update

Properties

Name Home

Icon dashboard

State ☒ Enabled

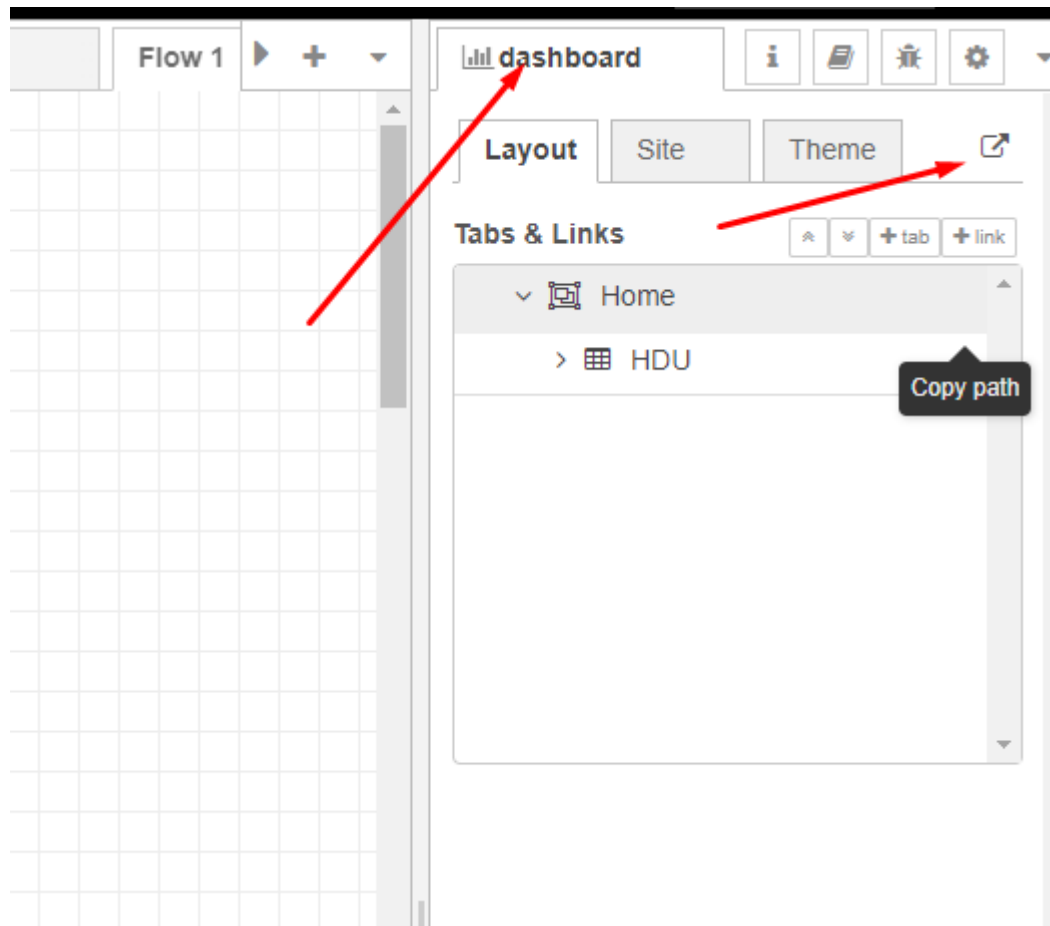
Nav. Menu ☒ Visible

The **Icon** field can be either a Material Design icon (e.g. 'check', 'close') or a Font Awesome icon (e.g. 'fa-fire'), or a Weather icon (e.g. 'wi-wu-sunny').

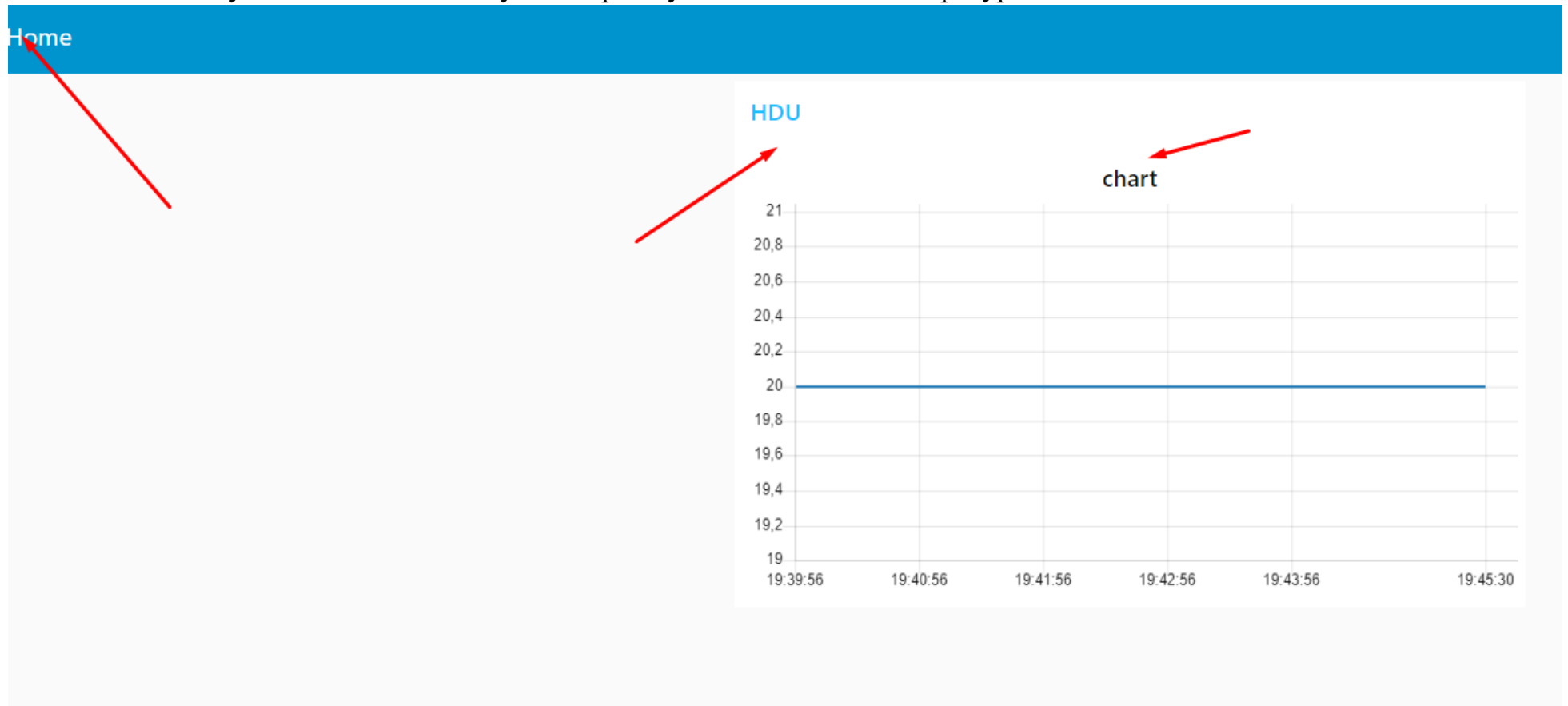
You can use the full set of google material icons if you add 'mi-' to the icon name. e.g. 'mi-videogame_asset'.

Теперь запустите свой флоу и проверьте панель инструментов на левой панели.

Щелкните значок панели инструментов, чтобы открыть панель инструментов в новом окне браузера.



Здесь вы можете увидеть свою линейную диаграмму со значением температуры.



Попробуйте изменить значение температуры и проверьте свою диаграмму.

Теперь давайте изменим вашу диаграмму, чтобы сделать ее более удобной для пользователя. Дважды щелкните по ноду диаграммы и откройте его свойства. Изменить метку с диаграммы на диаграмму температуры; измените свойства оси X, например, с 1 часа на 5 минут.

Edit chart node

Delete Cancel Done

Properties

Group [Home] HDU

Size auto

Label Temperature Chart

Type Line chart ☐ enlarge points

X-axis last 5 minute: OR 1000 points

X-axis Label HH:mm:ss ☐ as UTC

Y-axis min max

Legend None Interpolate linear

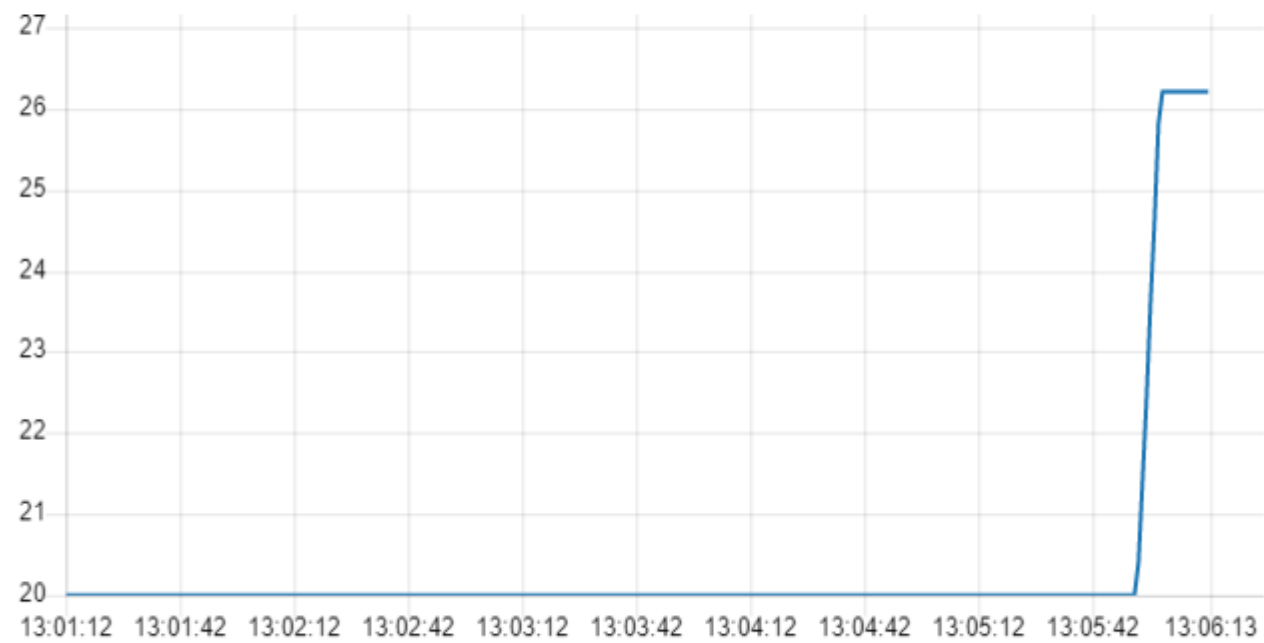
Series Colours

Blank label display this text before valid data arrives

Name Temperature

HDU

Temperature Chart



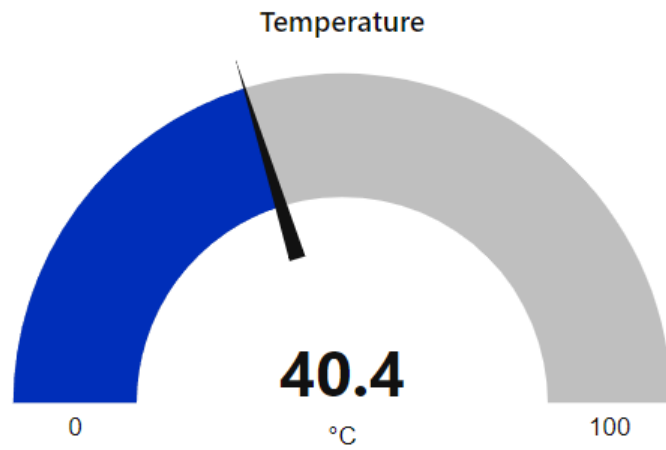
Теперь давайте добавим gauge визуализацию датчика. Измените диапазон от 0-10 до 0-100 и добавьте немного информации. Запустите ваш флоу и откройте панель инструментов.

The screenshot shows the Node-RED dashboard editor. On the left, a dashboard contains a 'gauge' widget. The main workspace shows a flow starting with a 'Sense HAT Sim' sensor (connected). The flow passes through a 'Temperature' function node, then a 'Temperature Threshold' node, and finally branches into 'Safe' and 'Critical' nodes. A 'Temperature Gauge' node is connected to the 'Temperature' function node. The 'Edit gauge node' panel is open on the right, showing the following properties:

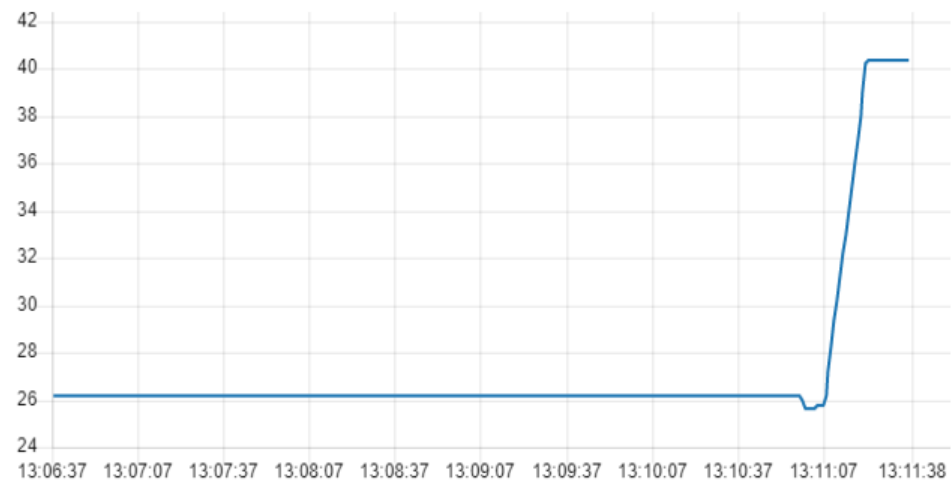
- Group:** [Home] HDU
- Size:** auto
- Type:** Gauge
- Label:** Temperature
- Value format:** {{value}}
- Units:** °C
- Range:** min 0, max 100
- Colour gradient:** Green, Blue, Red
- Sectors:** 0, optional, optional, 100
- Name:** Temperature Gauge

Red arrows point from the 'gauge' widget in the dashboard to the 'Temperature Gauge' node in the flow, and from the 'Done' button in the 'Edit gauge node' panel to the 'gauge' widget. Another red arrow points to the 'Range' field in the 'Edit gauge node' panel.

HDU



Temperature Chart



Вы также можете установить цветовой градиент для датчика, отображающего критические значения температуры.

The image shows a software interface for editing a gauge node. The dialog box is titled "Edit gauge node" and contains several configuration options. Red arrows highlight specific elements: the "Done" button, the "max" range field, the "Colour gradient" bar, and the "39" and "40" sector values.

Edit gauge node

Buttons: Delete, Cancel, Done

Properties

- Group: [Home] HDU
- Size: auto
- Type: Gauge
- Label: Temperature
- Value format: {{value}}
- Units: °C
- Range: min 0 max 100
- Colour gradient: [Green bar] [Red bar] [Dark red bar]
- Sectors: 0 ... 39 ... 40 ... 100
- Name: Temperature Gauge

Edit gauge node

Delete

Cancel

Done

⚙ Properties

⚙

📄

🖼

🏠 Group

[Home] HDU

▼

✎

📏 Size

auto

☰ Type

Gauge

▼

🏷 Label

Temperature

🏷 Value format

{{value}}

🏷 Units

°C

Range

min

0

max

100

Colour gradient

Sectors

0

...

39

...

40

...

100

🏷 Name

Temperature Gauge

ga

Adc

inte

The

val

defi

form

For

rou

app

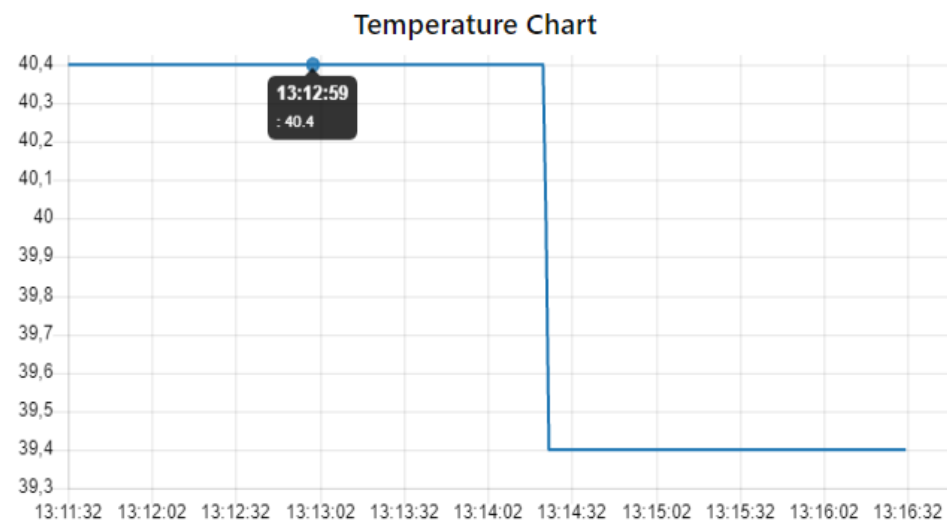
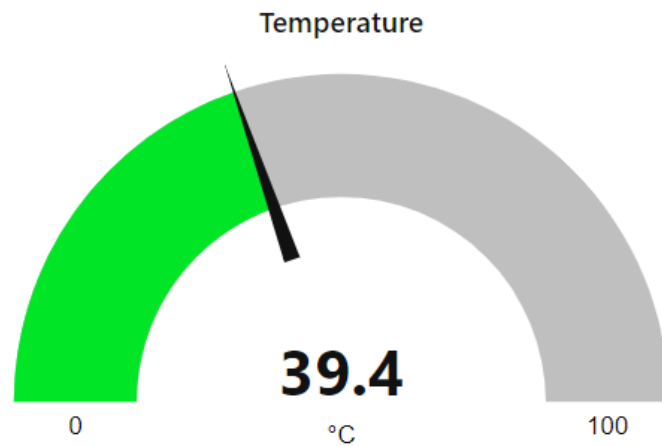
The

spe

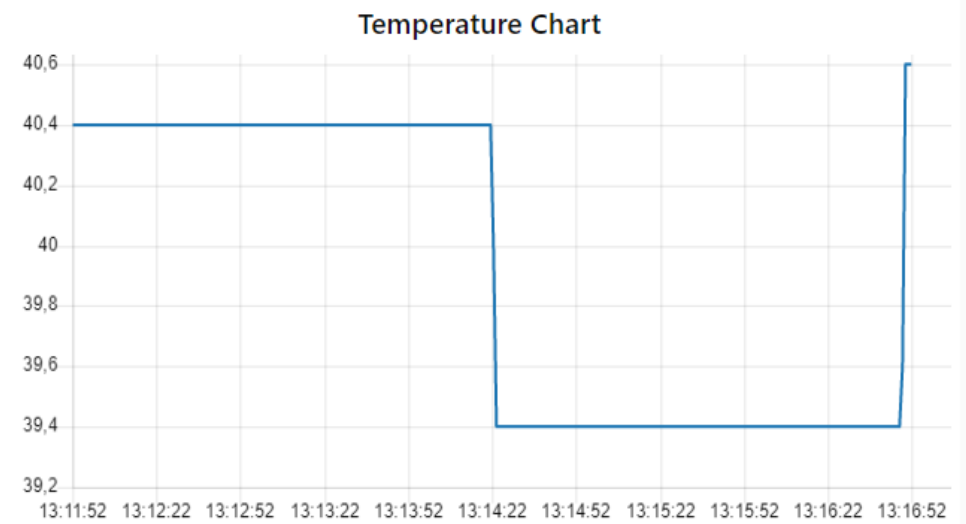
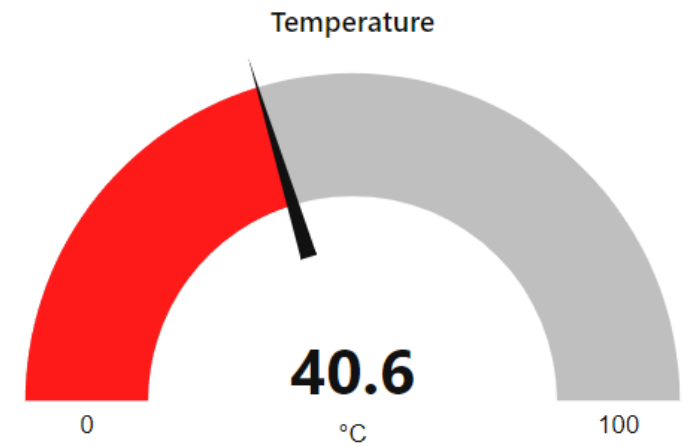
thei

(#rr

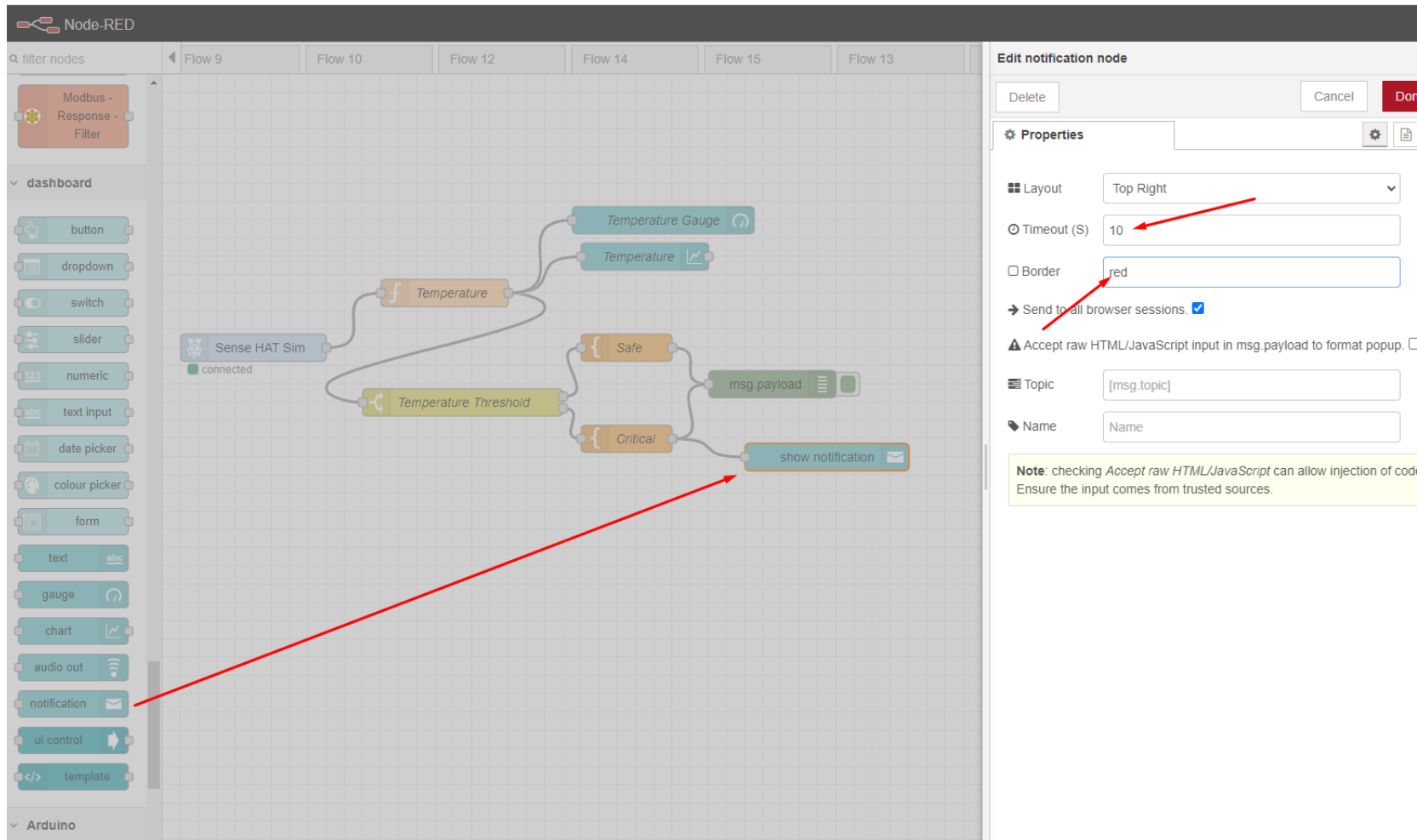
HDU



HDU



И, наконец, давайте добавим уведомление, если температура критическая. Добавьте notification node в свой рабочий процесс и свяжите его с critical template node. Установите желаемый тайм-аут и цвет границы. Запустите ваш флоу и откройте панель инструментов.



40.4 Warning - temperature critical !

HDU

