



SIEMENS



## Learn-/Training Document

Siemens Automation Cooperates with Education  
(SCE)

**TIA Portal Module 094-100**  
Node-RED with SIMATIC IOT2000

[siemens.com/sce](https://siemens.com/sce)

**SIEMENS**

Global Industry  
Partner of  
WorldSkills  
International



## Matching SCE Trainer Packages for this Learn-/Training Document

### Open Source Plattform

- **SIMATIC IOT2020 with Intel Quark x1000, 512 MB RAM, 1 x Ethernet, 1 x USB**  
Order no.: 124-4037 – orderable via RS Components [rs-components.com](https://www.rs-components.com)
- **SIMATIC IOT2040 with Intel Quark x1020 (+Secure Boot), 1 GB RAM, 2 x Ethernet, 2 x RS232/485, 1 x USB, RTC**  
Order no.: 6ES7647-0AA00-1YA2
- **SIMATIC IOT2000EDU S7 Software Controller runs with IOT2020 and IOT2040**  
Order no.: 6ES7671-0LE00-0YB0
- **SIMATIC IO-Shield: SIMATIC IOT2000 Input/Output Module with 5 DE, 2 DA, 2 AE, ARDUINO Shield for IOT2020/2040**  
Order no.: 6ES7647-0KA01-0AA2
- **3rd Party IO-Shield: IKHDS-Powershield for IOT2020/2040 with 6 DE, 5 DA (Relais), 1 DA (PWM), 2 AE, 1 AA**  
Order no.: 100301 – orderable via KAFTAN media UG [kaftan-media.com/iot2000](https://kaftan-media.com/iot2000)

Note that these trainer packages are replaced with successor packages when necessary.

An overview of the currently available SCE packages is available at: [siemens.com/sce/tp](https://siemens.com/sce/tp)

### Continued training

For regional Siemens SCE continued training, get in touch with your regional SCE contact

[siemens.com/sce/contact](https://siemens.com/sce/contact)

### Additional information regarding SCE

[siemens.com/sce](https://siemens.com/sce)

### Information regarding use

The SCE Learn-/Training Document for the integrated automation solution Totally Integrated Automation (TIA) was prepared for the program "Siemens Automation Cooperates with Education (SCE)" specifically for training purposes for public educational facilities and R&D institutions. Siemens does not guarantee the contents.

This document is only to be used for initial training on Siemens products/systems. This means it can be copied in whole or in part and given to trainees/students for use within the scope of their training/course of study. Disseminating or duplicating this document and sharing its content is permitted within public training and advanced training facilities for training purposes or as part of a course of study.

Exceptions require written consent from Siemens. Send all related requests to [scsupportfinder.i-ia@siemens.com](mailto:scsupportfinder.i-ia@siemens.com).

Offenders will be held liable. All rights including translation are reserved, particularly if a patent is granted or a utility model or design is registered.

Use for industrial customer courses is explicitly not permitted. We do not consent to commercial use of the Learn-/Training Document.

We wish to thank the TU Dresden, the Michael Dziallas Engineering Corporation and all other involved persons for their support during the preparation of this Learn-/Training Document.

# Table of contents

|       |   |    |
|-------|---|----|
| 1     | Goal .....                                  | 8  |
| 2     | Requirement .....                           | 8  |
| 3     | Required hardware and software .....        | 9  |
| 4     | Theory.....                                 | 10 |
| 4.1   | General information about Node-RED.....     | 10 |
| 4.2   | Node.js runtime environment.....            | 11 |
| 4.2.1 | npm: the Node.js package manager.....       | 11 |
| 5     | Task .....                                  | 12 |
| 6     | Planning .....                              | 12 |
| 7     | Structured step-by-step instructions .....  | 13 |
| 7.1   | Install or update Node-RED .....            | 13 |
| 7.2   | Manually start and stop Node-RED .....      | 14 |
| 7.3   | Automatically start and stop Node-RED ..... | 15 |
| 7.4   | Display the log file.....                   | 15 |
| 7.5   | Open the user interface .....               | 16 |
| 7.6   | Installation of new nodes .....             | 17 |
| 7.7   | Create a message in the flow .....          | 20 |
| 7.8   | Send messages to the cloud .....            | 23 |
| 7.9   | Checklist – step-by-step instructions ..... | 25 |
| 8     | Additional information .....                | 26 |

# Node-RED with SIMATIC IOT2000

## 1 Goal

The following pages show how to set up and program Node-RED on SIMATIC IOT2000 using the example image.

## 2 Requirement

This module is based on the module "SCE\_EN\_014-101 Hardware Configuration IOT2000". The hardware configuration up to and including section 4.3 must be completed to perform this module.

In addition, IOT2000 requires a working Internet connection. The easiest way to do this is to set the IOT to DHCP and connect it to a suitable router.

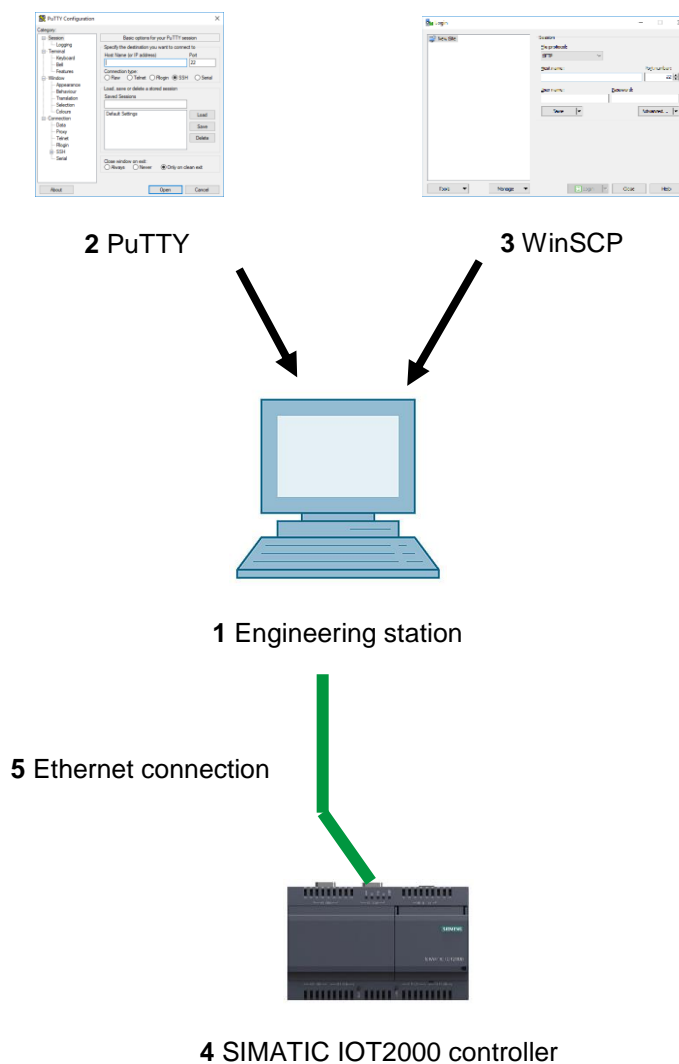
In the case of IOT2040, you can configure the **X1** interface statically (e.g. 192.168.0.1/24) and connect the **X2** interface to the router. This is preset to DHCP. The subnets for **X1** and **X2** must not overlap!

**Note:**

- *The IOT2000EDU Runtime, which may already be installed, must be disabled for this chapter!*

### 3 Required hardware and software

- 1 Engineering station: Requirements include hardware and operating system
- 2 Software for SSH Access, e.g. PuTTY
- 3 Software for SFTP/SCP File Transfer, e.g. WinSCP
- 4 SIMATIC IOT2000 Controller, e.g. IOT2040
- 5 Ethernet connection between engineering station and controller



## 4 Theory

### 4.1 General information about Node-RED

Node-RED is a free tool or development environment for interconnecting diverse hardware devices, APIs and online services. The software was originally developed by IBM and later released as open source software. Since then, it has been constantly further developed and is freely available to everyone.

The program offers a web interface which can be used for flow-based programming, similar to function block diagram (FBD) or ladder diagram (LAD) programming of Siemens controllers. The individual blocks available are called "nodes" here and are comparable to FCs or FBs. They offer inputs and outputs with which the individual nodes can be connected.

Data is transferred between the blocks in the form of messages. Each message consists of a title called a topic and a content called a payload. These messages are displayed as JSON (JavaScript Object Notation).

In addition to the standard nodes there is an active community that develops additional nodes and makes these freely available. The public library is available on the Node-RED website: [flows.nodered.org](https://flows.nodered.org)

Node-RED is written in JavaScript. It is possible to develop your own nodes. Documentation is available on the documentation page of the project: [nodered.org/docs/](https://nodered.org/docs/)

## 4.2 Node.js runtime environment

Node-RED uses node.js as the runtime environment. This JavaScript runtime environment is based on Chrome V8 JavaScript Engine and is also freely available. Node.js itself is lean and very efficiently structured. It can be used for various operating systems and is pre-installed in the example image for the Siemens IOT2000.

### 4.2.1 npm: the Node.js package manager

Node.js comes with its own package manager named npm. It can be used to easily install the desired nodes later.

The package manager can install packages both globally and locally. In a global installation, the package is available to the user everywhere. Locally installed packages, on the other hand, are stored in the current folder. This should be appropriately created beforehand.

In any case, the package can only be used by the current user. This has the advantage that npm does not need administrative rights to install packages. However, each user must install the required packages themselves.

As an alternative, you can create a folder, install the packages locally in that folder, and then share the folder with other users.



## 5 Task

In this chapter, the Node-RED service is set up and started with SIMATIC IOT2000 from the "SCE\_EN\_014-101\_Hardware Configuration IOT2000" chapter.

The Node-RED service is used to read the values of the analog input and upload them to the cloud.

## 6 Planning

The Node-RED service is set up on SIMATIC IOT2000 using the SSH client, PuTTY, via the command line.

In addition, some files have to be installed manually on IOT2000. This requires an SCP/SFTP connection, which is established with WinSCP support.

After establishing both connections, the **node-red** script must be copied to the **/etc/init.d/** folder on IOT2000 and marked as executable there. This script is included in the documentation archive for this document.

Next, you can start and stop Node-RED with this script.

As soon as Node-RED has been started, a flow that reads in and transmits the analog input can be programmed via the web interface.

The IBM Watson service is used as the cloud here for simplicity's sake. This does not require any registration and can be freely used. However, it only forwards data to the currently connected clients and does not store it.

## 7 Structured step-by-step instructions

You can find instructions on how to implement planning below. If you have advanced knowledge, the numbered steps are sufficient. Otherwise, it is recommended that you follow the individual steps of the instructions.

### 7.1 Install or update Node-RED

Node.js, the corresponding package manager npm and Node-RED were pre-installed on the example image from Siemens. However, the versions are relatively old, so npm and Node-RED can be updated beforehand.

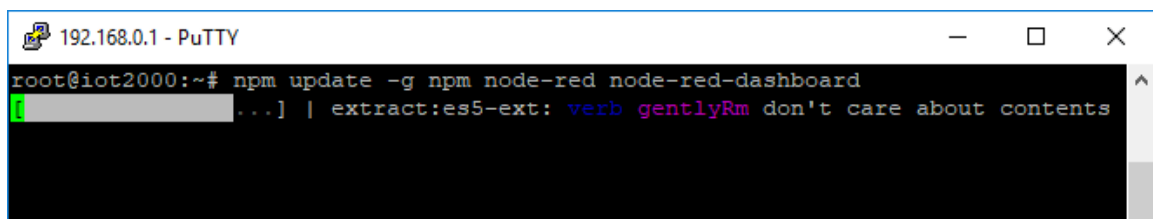
**Note:**

- *SIMATIC IOT must have a functioning Internet connection for this. You can achieve this with the IOT2040, for example, by connecting the X2 interface to a router*

To update the node.js modules, use the command line of the IOT.

→ Update the installed node.js modules with the following command:

```
npm install -g npm node-red node-red-dashboard
```

A screenshot of a PuTTY terminal window titled '192.168.0.1 - PuTTY'. The terminal shows a command prompt 'root@iot2000:~#' followed by the command 'npm update -g npm node-red node-red-dashboard'. The output of the command is partially visible, showing '[...]' and some text about 'extract:es5-ext: verb gentlyRm don't care about contents'. The terminal has a black background with white and green text.

**Note:**

- *This operation takes a relatively long time!*

## 7.2 Manually start and stop Node-RED

The software is started via the command line. Note that Node-RED stores its data in the "node-red" folder in the home directory of the user.

To run Node-RED in the background, the **node-red** script must be copied from the documentation archive of the IOT.

→ Copy the **node-red** script from the documentation archive to the **/etc/init.d/** folder.

→ Make the file executable:

```
chmod +x /etc/init.d/node-red
```

```
root@iot2000:~# chmod +x /etc/init.d/node-red
root@iot2000:~#
```

→ Start Node-RED by calling the script with the **start** parameter:

```
/etc/init.d/node-red start
```

```
root@iot2000:~# /etc/init.d/node-red start
Starting Node-RED: logging to /var/log/node-red.log
root@iot2000:~#
```

### Note:

- This script stores all output of Node-RED in the file `/var/log/node-red.log`, for more information see "[Displaying the Log File](#)".

→ You can stop Node-RED by calling the script with the **stop** parameter:

```
/etc/init.d/node-red stop
```

```
root@iot2000:~# /etc/init.d/node-red stop
Stopping Node-RED: .....done
root@iot2000:~#
```

### Note:

- Do not use the `iot2000setup` tool to start Node-RED. This tool prevents you from viewing the event log of Node-RED!

## 7.3 Automatically start and stop Node-RED

The Linux system is able to execute the script from the previous section automatically at startup.

→ Add the script to the autostart:

```
update-rc.d node-red defaults
```

```
root@iot2000:~# update-rc.d node-red defaults
Adding system startup for /etc/init.d/node-red.
root@iot2000:~#
```

→ You can remove the script from the autostart with the following command:

```
update-rc.d -f node-red remove
```

```
root@iot2000:~# update-rc.d -f node-red remove
update-rc.d: /etc/init.d/node-red exists during rc.d purge (continuing)
Removing any system startup links for node-red ...
/etc/rc0.d/K20node-red
/etc/rc1.d/K20node-red
/etc/rc2.d/S20node-red
/etc/rc3.d/S20node-red
/etc/rc4.d/S20node-red
/etc/rc5.d/S20node-red
/etc/rc6.d/K20node-red
root@iot2000:~#
```

## 7.4 Display the log file

With the script installed under "[Manually start and stop Node-RED](#)", Node-RED was started so that program events such as errors or warnings are written to the file **/var/log/node-red.log**.

This file can be monitored via the SSH connection and the "tail" program.

→ Follow the changes in the file **/var/log/node-red.log** with **tail**:

```
tail -f /var/log/node-red.log
```

```
root@iot2000:~# tail -f /var/log/node-red.log
22 Mar 13:06:03 - [info] Dashboard version 2.3.11 started at /ui
22 Mar 13:06:05 - [warn] -----
22 Mar 13:06:05 - [warn] [rpi-gpio] Info : Ignoring Raspberry Pi specific node
22 Mar 13:06:05 - [warn] -----
22 Mar 13:06:05 - [info] Settings file   : /home/root/.node-red/settings.js
22 Mar 13:06:05 - [info] User directory  : /home/root/.node-red
22 Mar 13:06:05 - [info] Flows file     : /home/root/.node-red/flows_iot2000.js
22 Mar 13:06:05 - [info] Server now running at http://127.0.0.1:1880/
22 Mar 13:06:05 - [info] Starting flows
22 Mar 13:06:05 - [info] Started flows
```



### Note:

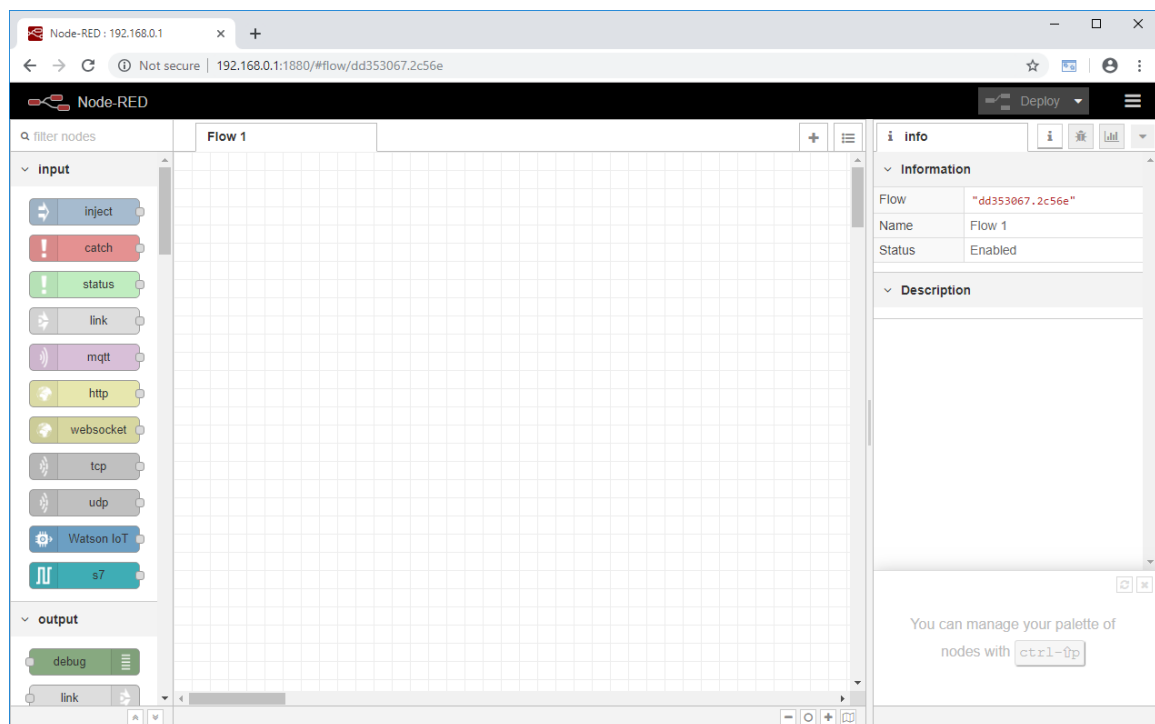
- The **-f** option specifies that tail should follow the file. This means that the file is monitored and changes are shown immediately. To exit tail and return to the command prompt, use the key combination **CTRL+C**.

## 7.5 Open the user interface

Once Node-RED has successfully started (the "Server now running at ..." line should be displayed in the log), the user interface can be opened. This is provided as a web service and can therefore be accessed via any JavaScript-enabled web browser.

The interface can be accessed unencrypted with the browser via the IP with SIMATIC IOT2000 and port 1880, e.g. <http://192.168.0.1:1880>

- Start your browser and open the development environment.  
(→ <http://192.168.0.1:1880/>)
- You can see all available nodes and blocks on the left. They are sorted by type and package and can be filtered via the search field (→ [filter nodes](#)).
- The editor is in the middle. It can be used to program the nodes (also referred to as "flows"). This is similar to the FBD Editor in the TIA Portal.
- On the right side you will find helpful information on the currently selected block in the **info** (  ) tab. The **debug** (  ) tab displays current debug information as well as any error messages from blocks.



## 7.6 Installation of new nodes

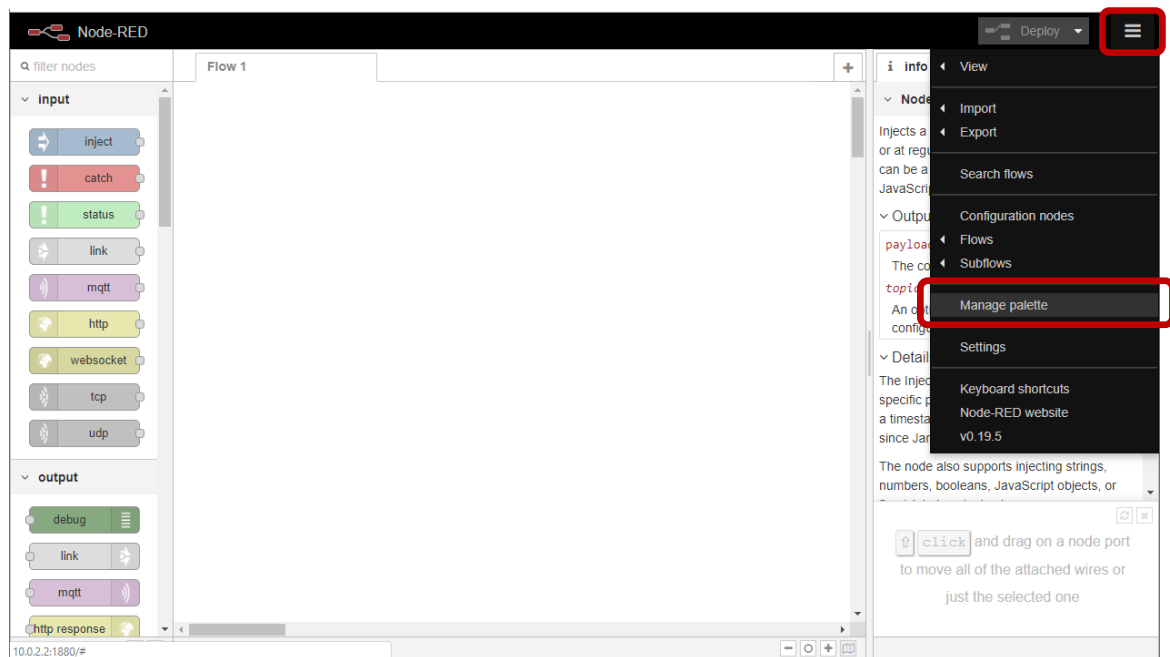
Communicating with the Watson IoT service requires nodes that are not included in the standard installation. These nodes can be installed directly via the web interface.

**Note:**

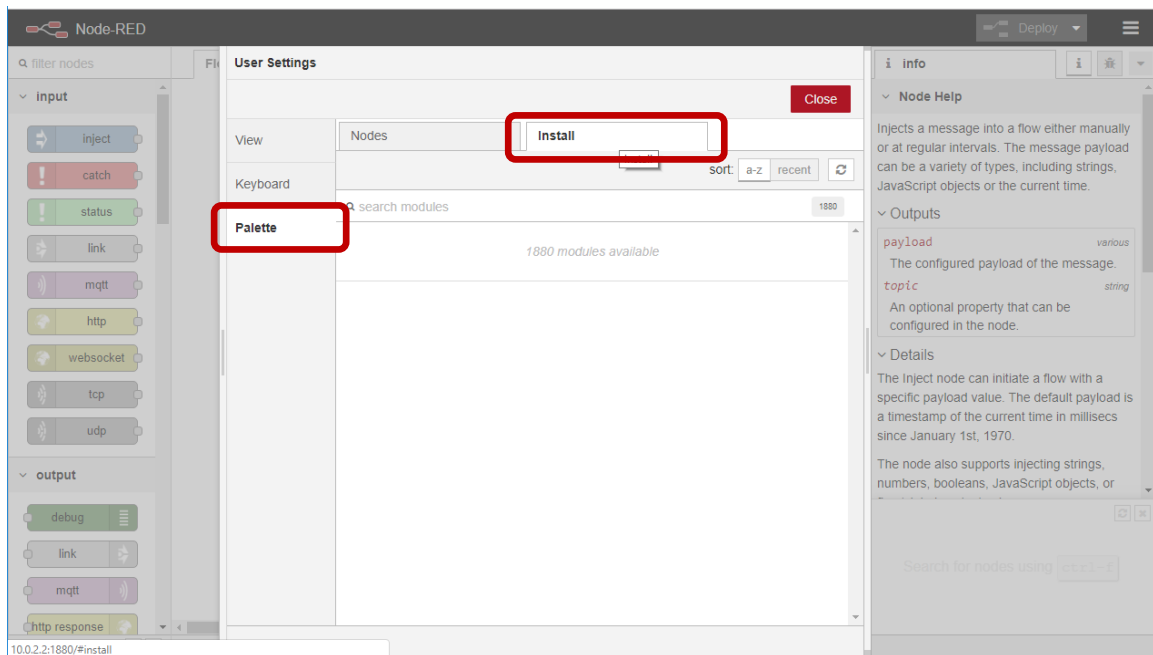
- *SIMATIC IOT must have a functioning Internet connection for this. You can achieve this with the IOT2040, for example, by connecting the X2 interface to a router.*

The installed packages are stored in the Node-RED user directory. By default, this is the ".node-red" folder in the home directory of the current user running Node-RED.

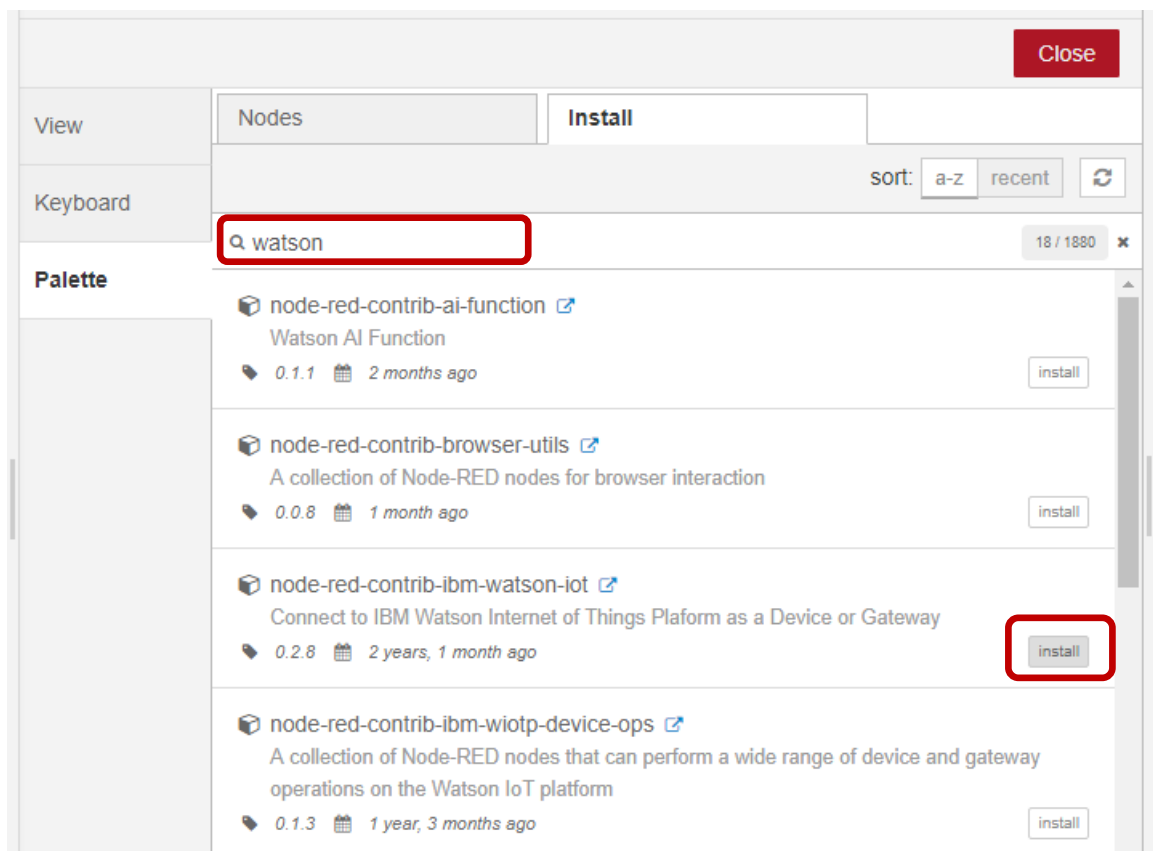
→ Open the menu and select **Manage palette**.



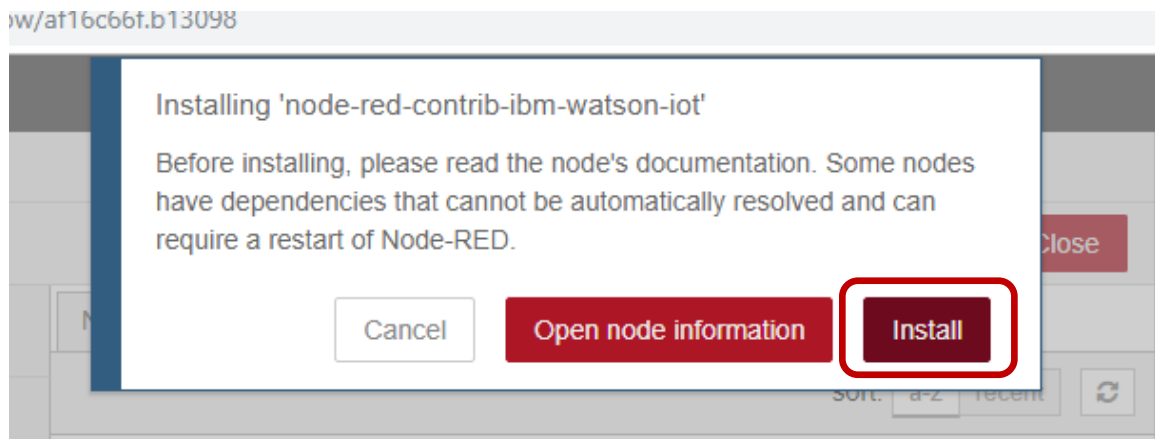
→ Now select the **Install** tab.



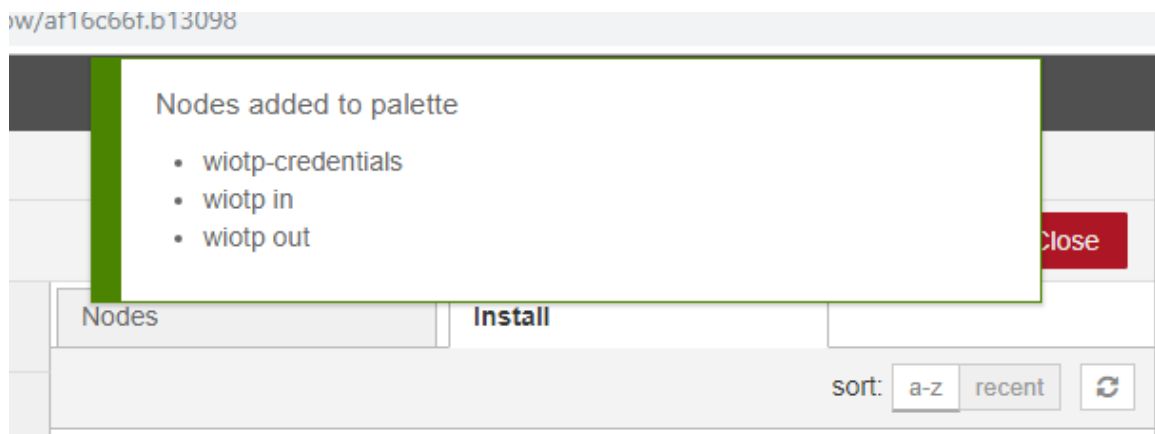
→ In the **search modules** field, search for the term **watson** and install the required **node-red-contrib-ibm-watson-iot** module by clicking **install**.



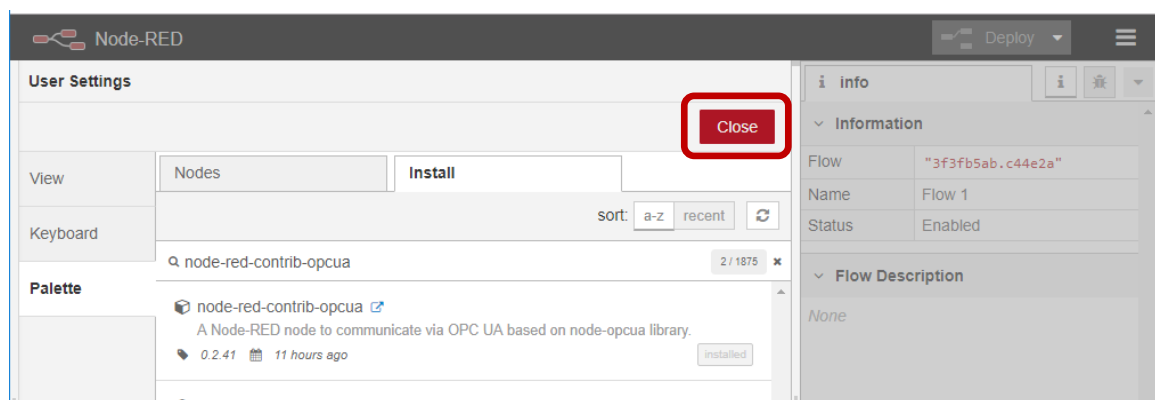
→ Confirm the installation of the node by clicking **Install**.



→ After a while, the installation should be confirmed.



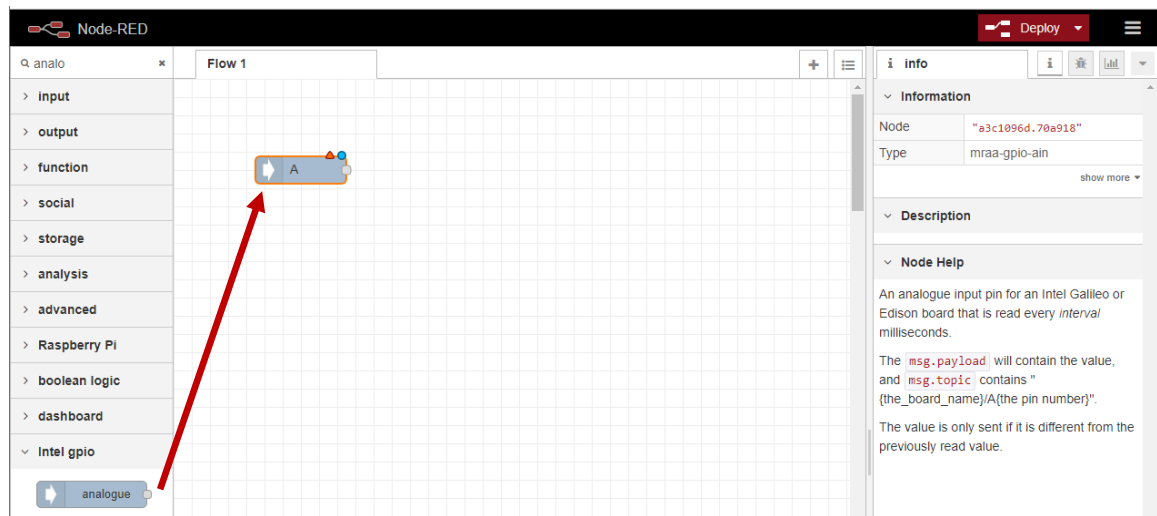
→ Close the palette by clicking **Close**.



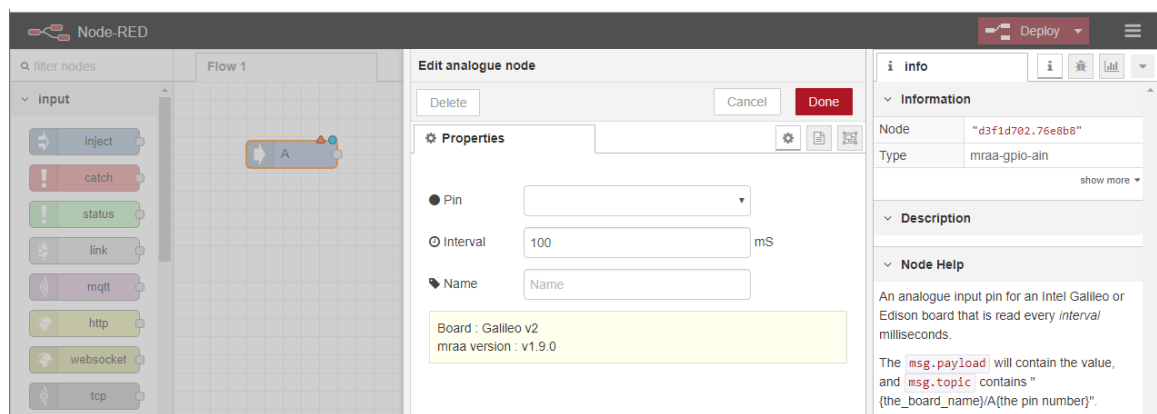


## 7.7 Create a message in the flow

→ Drag the **analogue** node from the **Intel\_gpio** category into the editor.



→ Double-click on the inserted node to open its properties.



- Under Pin, select the analog input on the Arduino UNO R3 header. This example uses the Siemens IOT2000 Input/Output Module. Input U0 is connected to pin A0 there. If you are using a different shield, refer to the manual of the module for the correct pin. (→ Pin: A0).
- Under Name, enter a name for this node: (→ Name: U0).
- Confirm the changes with Done: (→ Done).

**Edit analogue node**

Delete Cancel Done

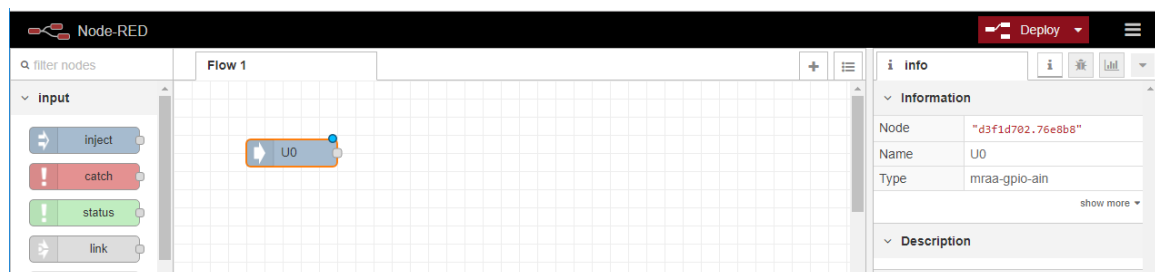
**Properties**

Pin A0

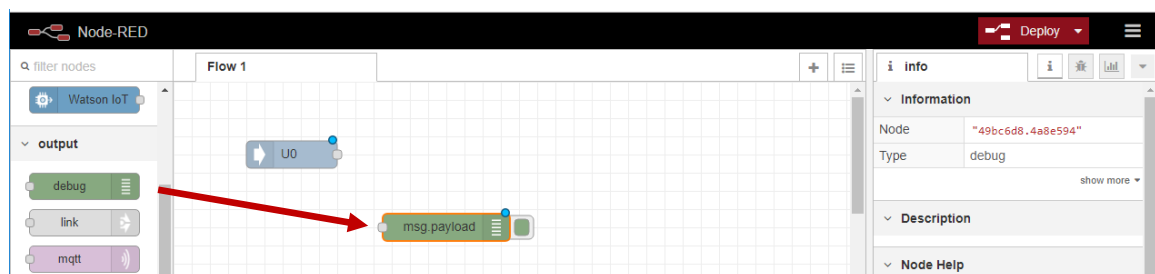
Interval 100 ms

Name U0

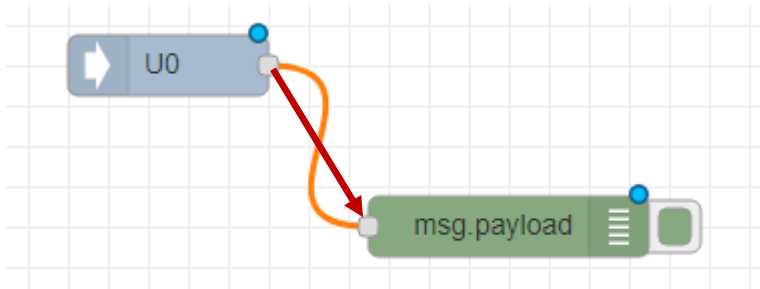
Board : Galileo v2  
mraa version : v1.9.0



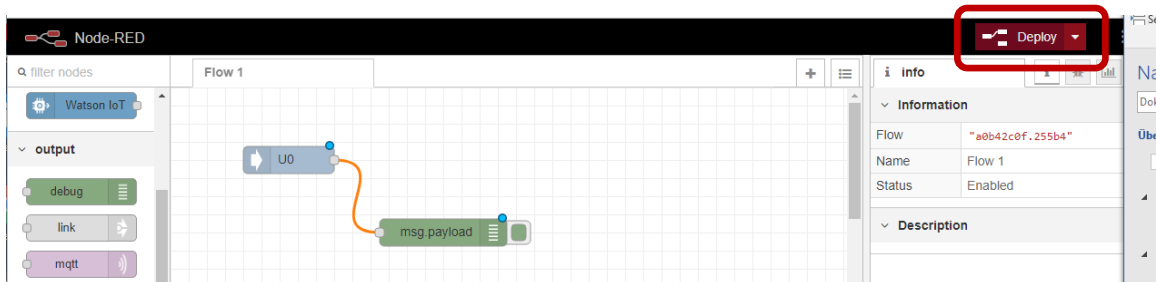
- Now drag a **debug** node from the **output** section into the editor.




→ Connect the output of the **U0** node to the input of the **debug** node by clicking on the output and pulling it to the input:



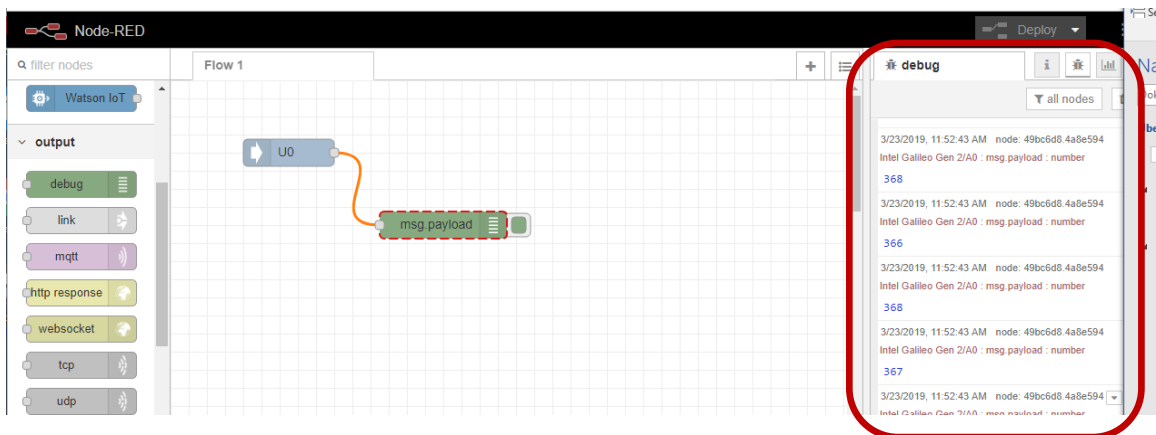
→ Now this simple flow can be activated via the **Deploy** button: (→ Deploy)



→ Switch to the debug view on the right sidebar: (→ )



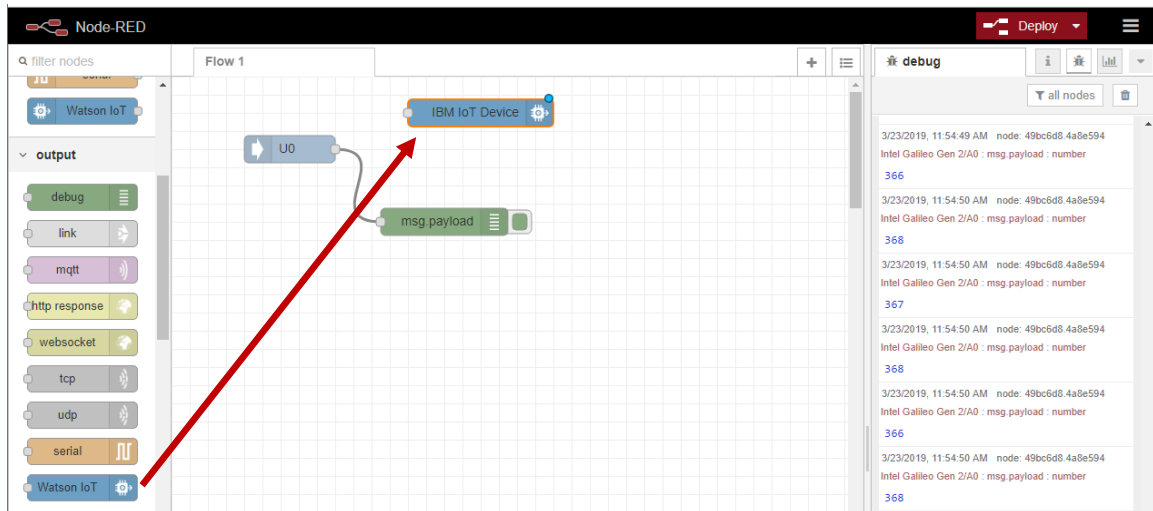
→ The **debug** tab should now show messages with the current analog value.



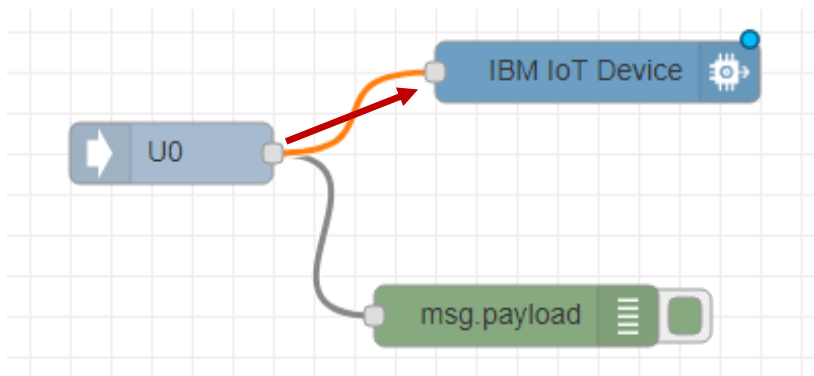
## 7.8 Send messages to the cloud

In the next step, the analog values just read will be sent to the debug output as well as to the Watson Cloud of IBM.

→ Drag the **Watson IoT Node** from the **outputs** into the editor:

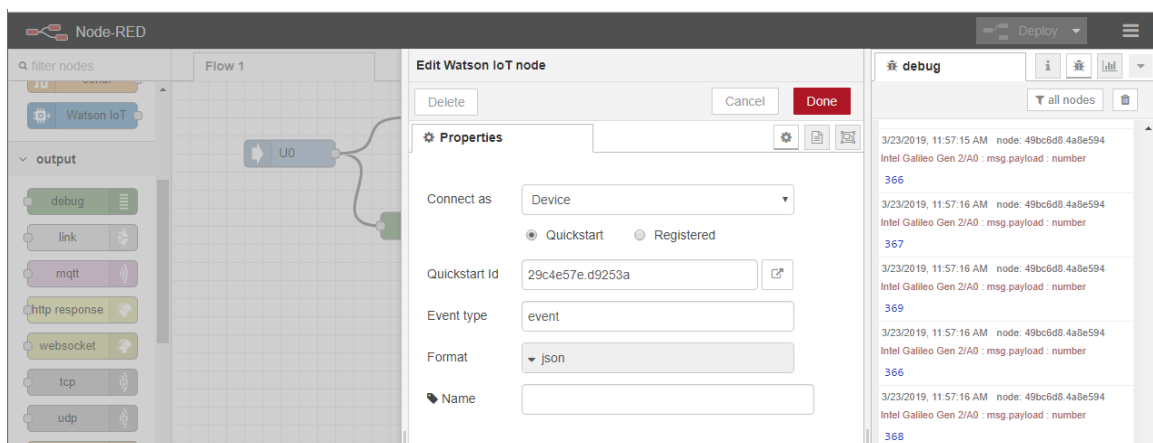



→ Establish a connection between the **U0** node and the **Watson IoT** node.



→ Activate the changes by clicking **Deploy**: (→ Deploy).

→ Double-click on the **Watson IoT** node to open its properties:



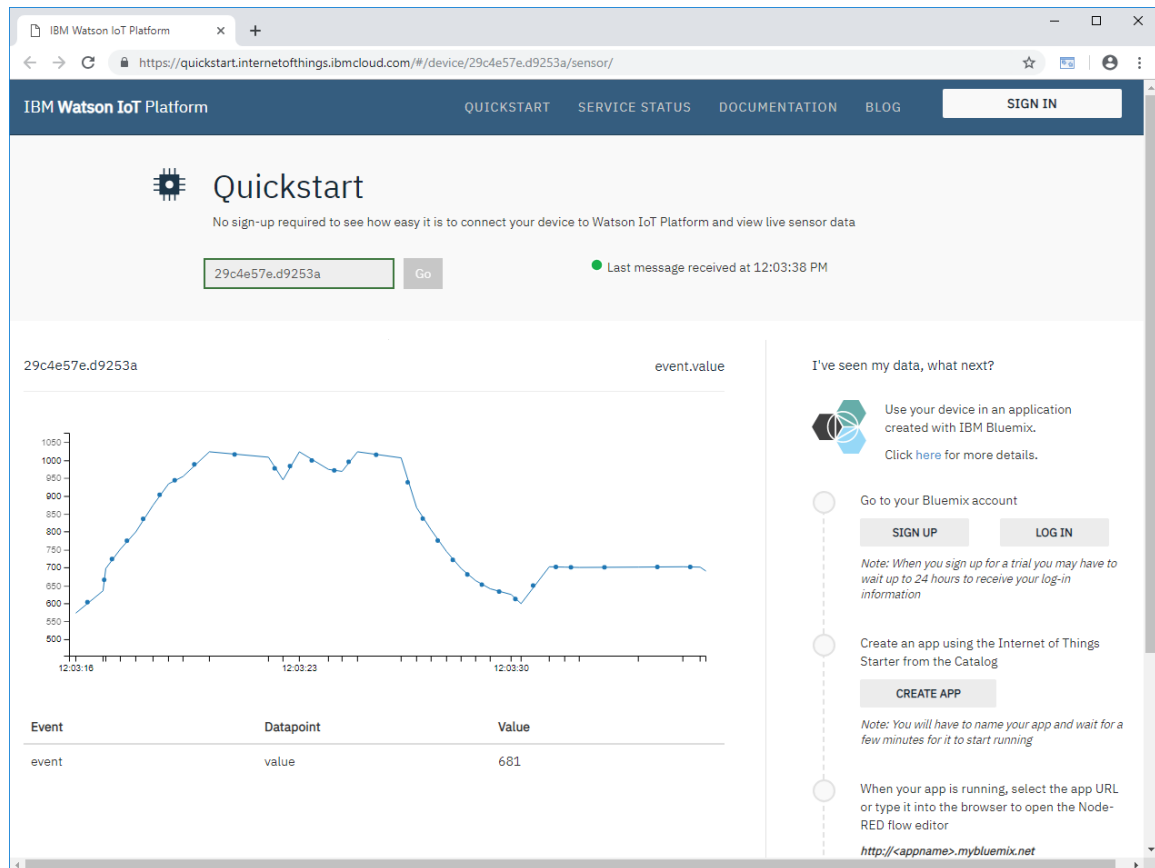
→ Click the button next to **Quickstart Id:** (→ )

Quickstart Id

29c4e57e.d9253a



→ A new browser window opens with the Watson IoT Platform. The measured values should now be displayed as a graph in this window. If necessary, you must accept the terms of use beforehand.



### Note:

- *Watson will only show changes in measured values. If the analog value does not change, nothing may be displayed here!*

## 7.9 Checklist – step-by-step instructions

The following checklist helps trainees/students to independently check whether all steps of the step-by-step instruction have been carefully completed and enables them to successfully complete the module on their own.

| No. | Description   | Checked |
|-----|---|---------|
| 1   | Start script "node-red" installed                     |         |
| 2   | Node-RED started                                      |         |
| 3   | Start procedure checked with the help of the log file |         |
| 4   | Analog value transferred to Watson node               |         |
| 5   | Deploy successful                                     |         |
| 6   | Analog values visible in the Watson Cloud             |         |

Table 1.7

## 8 Additional information

More information for further practice and consolidation is available as orientation, for example: Getting Started, videos, tutorials, apps, manuals, programming guidelines and trial software/firmware, under the following link:

[siemens.com/sce](https://www.siemens.com/sce)

### Preview "Additional information"

Getting Started, Videos, Tutorials, Apps, Manuals, Trial-SW/Firmware

- TIA Portal Videos
- TIA Portal Tutorial Center
- Getting Started
- Programming Guideline
- Easy Entry in SIMATIC S7-1200
- Download Trial Software/Firmware
- Technical Documentation SIMATIC Controller
- Industry Online Support App
- TIA Portal, SIMATIC S7-1200/1500 Overview
- TIA Portal Website
- SIMATIC S7-1200 Website
- SIMATIC S7-1500 Website

## Further Information

Siemens Automation Cooperates with Education  
**[siemens.com/sce](https://www.siemens.com/sce)**

SCE Learn-/Training Documents  
**[siemens.com/sce/documents](https://www.siemens.com/sce/documents)**

SCE Trainer Packages  
**[siemens.com/sce/tp](https://www.siemens.com/sce/tp)**

SCE Contact Partners  
**[siemens.com/sce/contact](https://www.siemens.com/sce/contact)**

Digital Enterprise  
**[siemens.com/digital-enterprise](https://www.siemens.com/digital-enterprise)**

Industrie 4.0  
**[siemens.com/future-of-manufacturing](https://www.siemens.com/future-of-manufacturing)**

Totally Integrated Automation (TIA)  
**[siemens.com/tia](https://www.siemens.com/tia)**

TIA Portal  
**[siemens.com/tia-portal](https://www.siemens.com/tia-portal)**

SIMATIC Controller  
**[siemens.com/controller](https://www.siemens.com/controller)**

SIMATIC Technical Documentation  
**[siemens.com/simatic-docu](https://www.siemens.com/simatic-docu)**

Industry Online Support  
**[support.industry.siemens.com](https://support.industry.siemens.com)**

Product catalogue and online ordering system Industry Mall  
**[mall.industry.siemens.com](https://mall.industry.siemens.com)**

Siemens  
Digital Industries, FA  
P.O. Box 4848  
90026 Nuremberg  
Germany

Subject to change and errors  
© Siemens 2019

**[siemens.com/sce](https://www.siemens.com/sce)**