

CHAPTER 1. Internet of Things

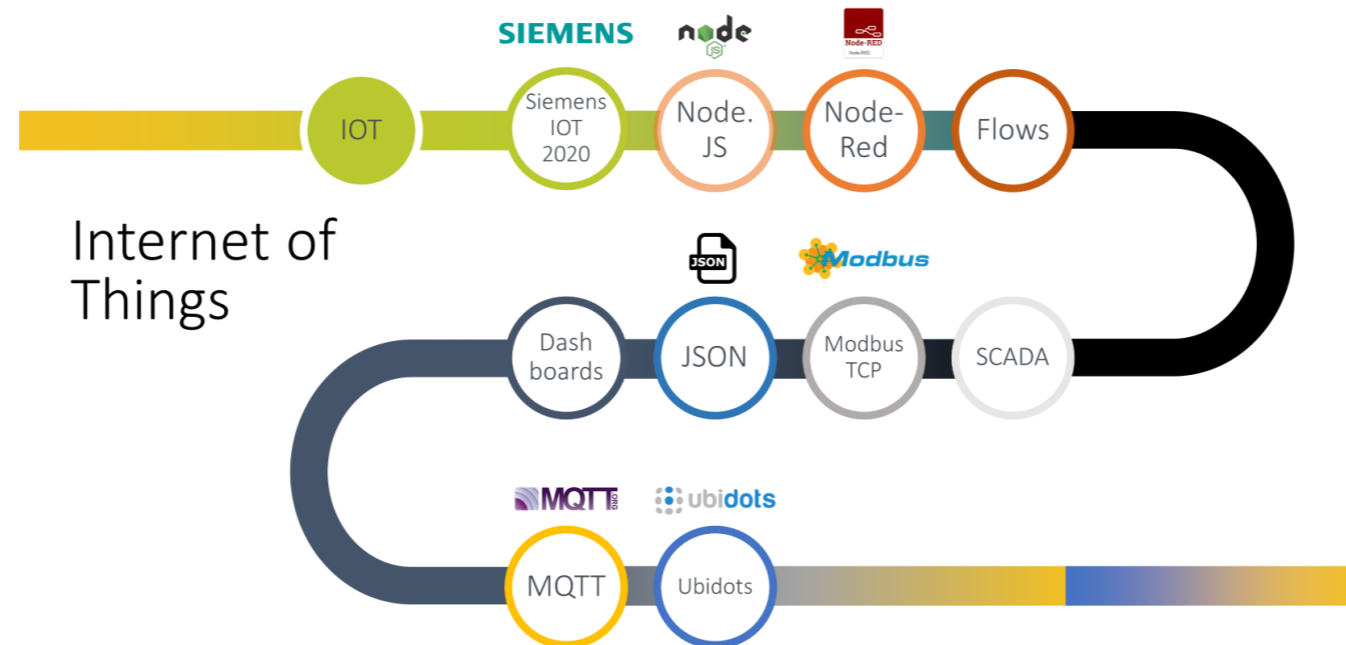
v.1.2 SEPTEMBER 2024



Ricardo Moraleda Gareta
[Director departamento
de software de GDO Software]

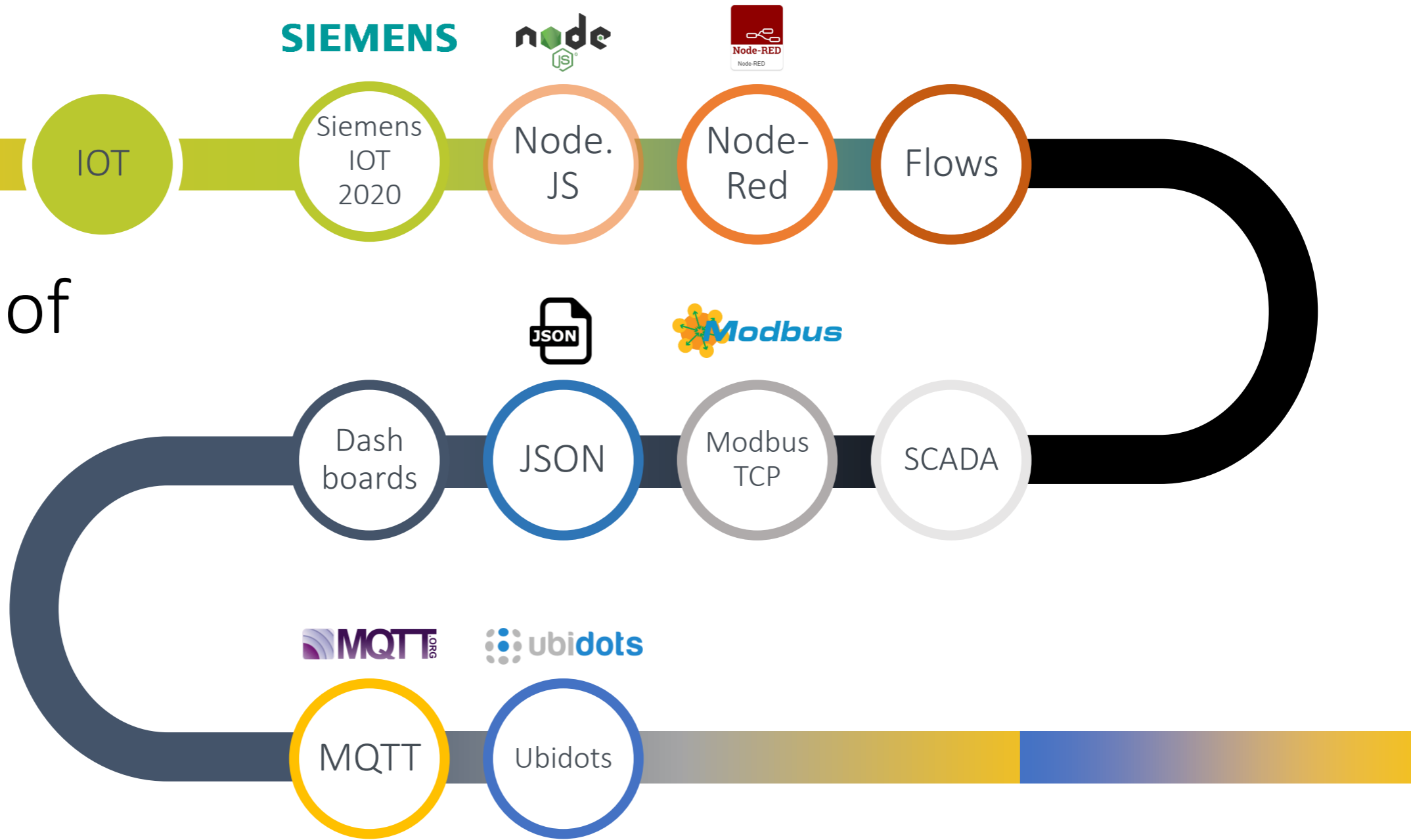


Melvin Francis Stephen
[Software Engineer
BE.services GmbH]



Internet of Things

v.1.2 SEPTEMBER 2024



IOT Device



SIMATIC IOT 2020 / 2040 **SIEMENS**

Ethernet Gateway manufactured by Siemens and based on Arduino.

Operating Systems: Yocto Linux 2.1

5 ED of 24 V & 2 SD, 2 EA between 0 & 10 V or between 0 and 20 mA

Programmable with high level languages (Java, C++, Python, JavaScript: Node.js / Node-RED)

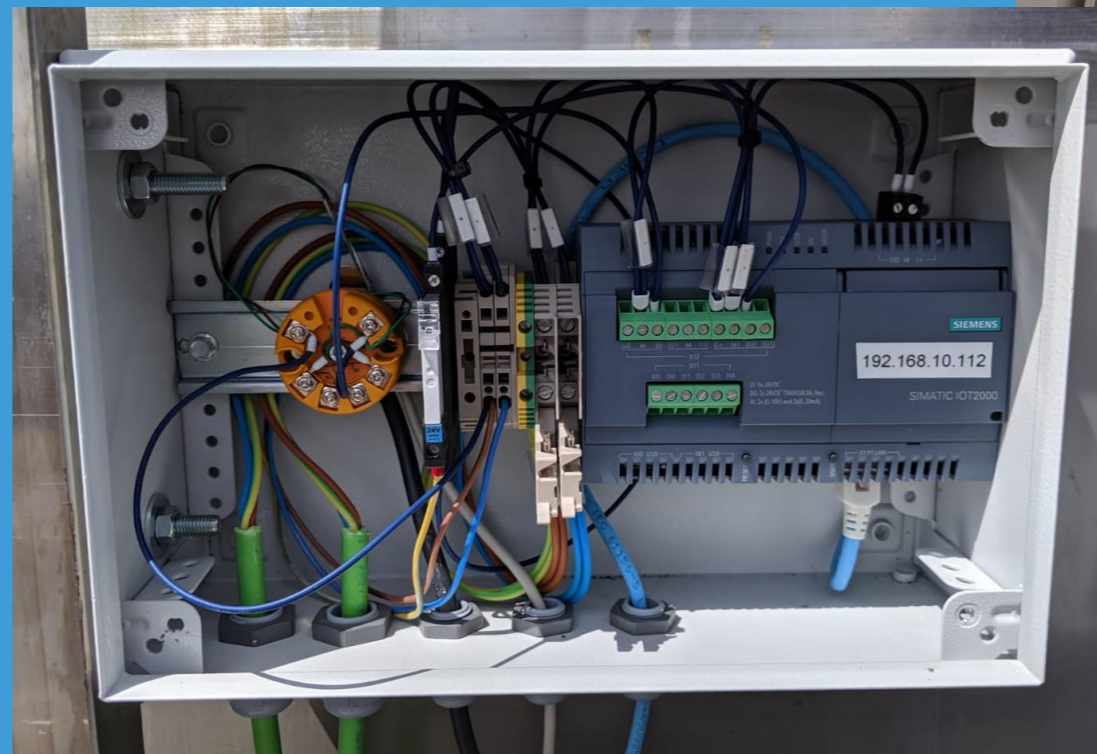
For more information on how to install-and-configure the SD and software, follow the link below (JANUARY 2018):
<https://drive.google.com/file/d/18NCgvOZVMuQlij5iruu0gIL-xa4y5eON/view>

IOT in real installation - 2020

IOT for each wine tank
controlling:

- PT100 temperature probe
(4-20 mA)
- Cold valve relay

Installations of 24 and 64
IOTs in an Ethernet Network





PC Modules



Node.js



Runtime environment for JavaScript

Step 1. Install Node.js

The first thing we are going to do is install Node.js. To do this, we will download the Windows version from its website <https://nodejs.org/en> . In this example, we will download and install the LTS 10.15.2 version.

To do this, again from the command prompt, we will execute: `npm install ; npm --version`

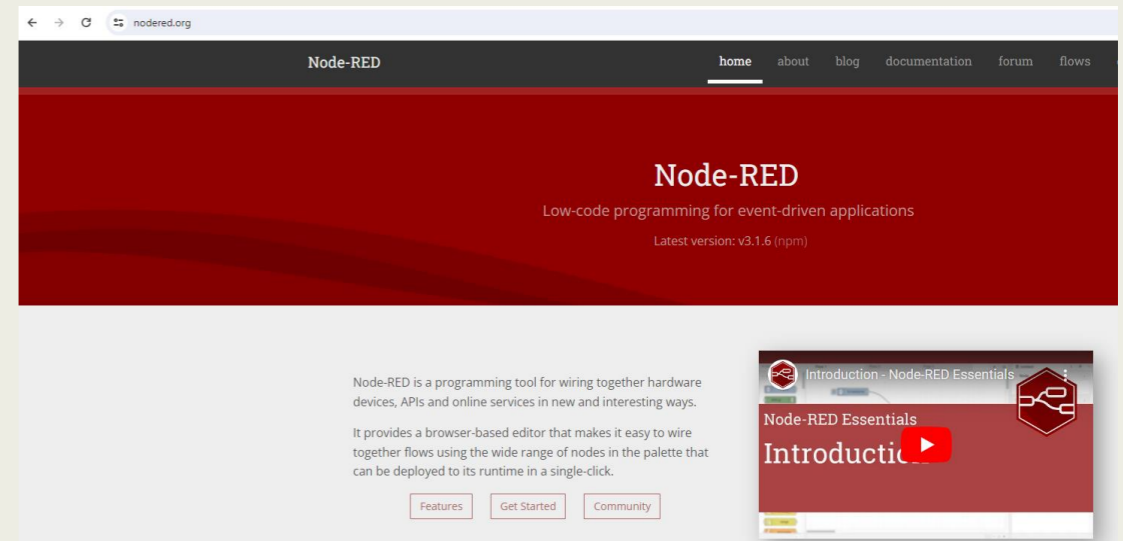
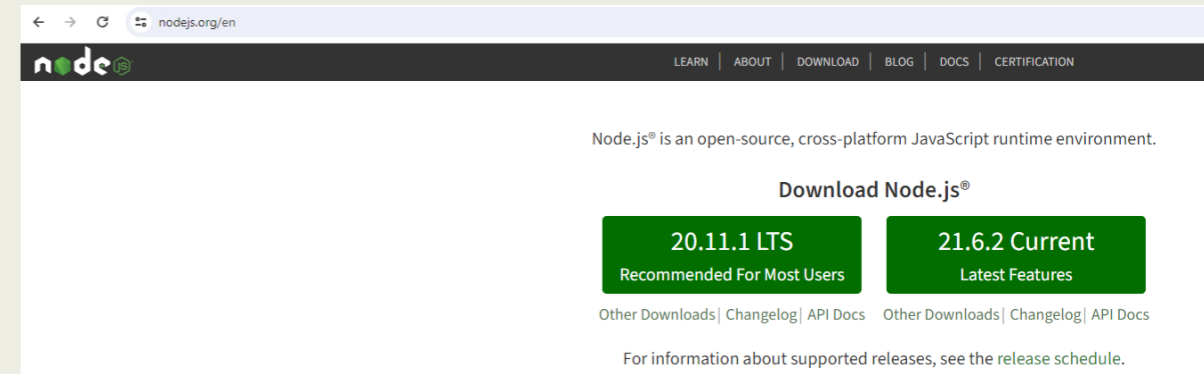
Node-Red



Step 2. Install Node-Red

Once node.js is installed, we will install Node-Red

To do this, again from the command prompt, we will execute: `npm install -g --unsafe-perm node-red`





Node-Red



Node-Red

Flow-based development tool for visual programming originally developed by IBM to connect hardware devices, APIs, and online services as part of the Internet of Things.

http://localhost:1880

The screenshot displays the Node-RED web interface in a browser window. The address bar shows the URL `localhost:1880/#flow/e8c35ae7.b7d968`. The interface includes a left-hand palette with 'input' and 'output' categories, a central workspace with a flow diagram, and a terminal window at the bottom.

The flow diagram in the workspace shows a sequence of nodes: an inject node with the text 'Bienvenida!Hola mundo!', followed by two function nodes labeled 'Sumando A' and 'Sumando B', and a final function node labeled 'Resultado'. Below this, there is a start node with the text `start({brightness: 0})` connected to an 'increment' node (checked), which is connected to an 'Escritura' node. To the right, there is a 'Lectura' node connected to a 'msg payload' node. Further right, there is a node for '/1.6/devices/plc_1200/brightness' connected to another 'Lectura' node.

The terminal window at the bottom shows the following log output:

```

node-red
28 Aug 14:59:12 - [info]
Welcome to Node-RED
-----
28 Aug 14:59:12 - [info] Node-RED version: v0.20.7
28 Aug 14:59:12 - [info] Node.js version: v10.16.3
28 Aug 14:59:12 - [info] Windows_NT 10.0.18362 x64 LE
28 Aug 14:59:15 - [info] Loading palette nodes
28 Aug 14:59:20 - [warn] rpi-gpio : Raspberry Pi specific node set inactive
28 Aug 14:59:21 - [info] Settings file : \Users\r.moraleda\.node-red\settings.js
28 Aug 14:59:21 - [info] Context store : 'default' [module=memory]
28 Aug 14:59:21 - [info] User directory : \Users\r.moraleda\.node-red
28 Aug 14:59:21 - [warn] Projects disabled : editorTheme.projects.enabled=false
28 Aug 14:59:21 - [info] Flows file : \Users\r.moraleda\.node-red\flows_GD042.json
28 Aug 14:59:21 - [info] Server now running at http://127.0.0.1:1880/
28 Aug 14:59:21 - [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.

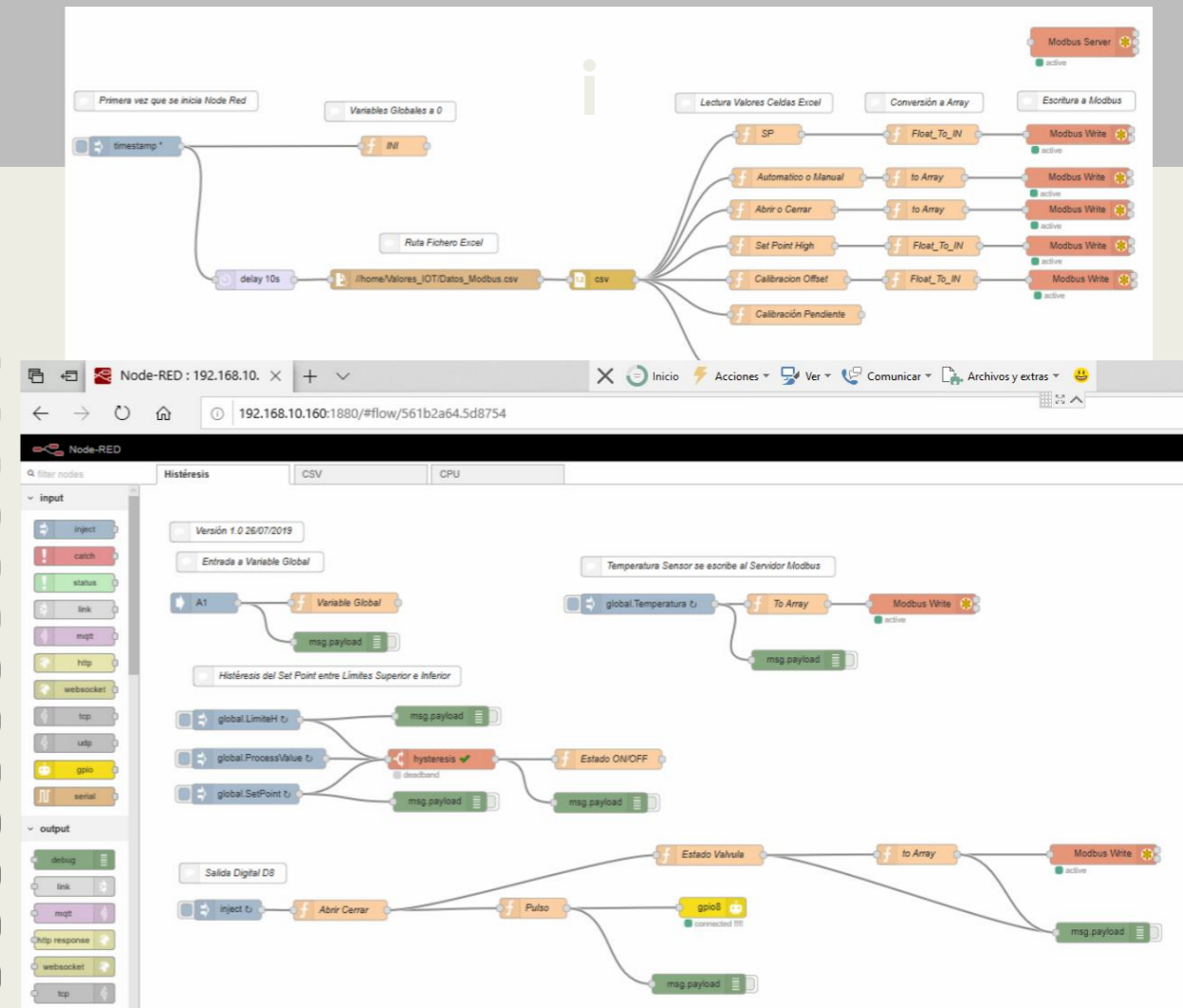
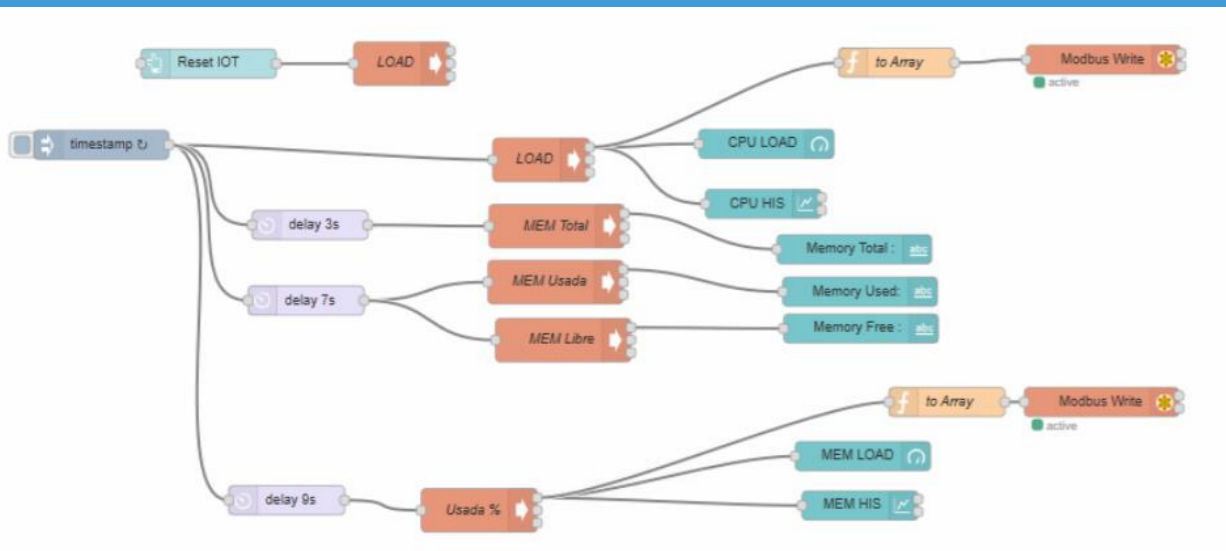
You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----
28 Aug 14:59:21 - [info] Starting flows
28 Aug 14:59:21 - [info] Started flows

```

Node-Red (flows)

Flow examples

- Process control
- Writing to local CSV file
- Publishing to dashboard





Node-Red (Dashboard)



<http://localhost:1880/ui>

Estado del Tanque **Automático**

Consigna en Manual **Abrir**

Temperatura Tanque



SetPoint SP **SP 17.50**

Límite Superior **SPH 1.0**

Valor Actual Calibración Pendiente **0.98**

Valor Actual Calibración Offset **-1.7**

Histórico Tanque



Valor Calibración Pendiente

CPU_IOT

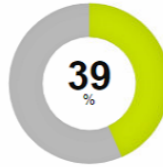
RESET IOT

Memory Total : 491.376 MB

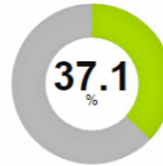
Memory Used: 182.504 MB

Memory Free : 309 MB

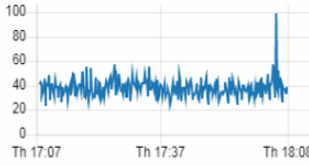
CPU LOAD



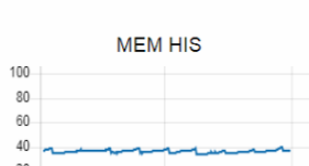
Memoria Usada



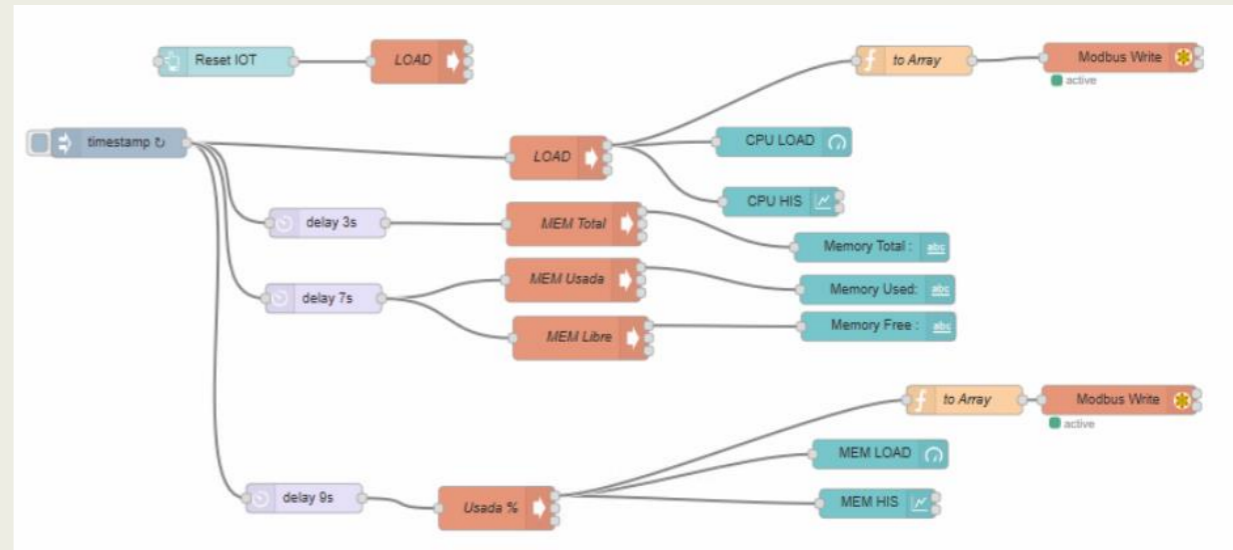
CPU HIS



MEM HIS



Flow





Node-Red (JSON)



JSON format flows



Acronym for JavaScript Object Notation, "JavaScript object notation") is a plain text format for exchanging data.

Exporting flows

Export nodes

Export to clipboard selected nodes current flow all flows

```

[{"id":"f62aefc5.d09ab","type":"dsm","z":"e8c35ae7.b7d968","name":"increment","sm_config":{"currentState":"stopped","states":{"stopped":{"start":"started"},"started":{"run":"running","end":"stopped"},"running":{"run":"run"}},\n

```

- Objetos: Son colecciones no ordenadas de pares de la forma <nombre>:<valor> separados por comas y puestas entre llaves. El nombre tiene que ser una cadena y entre ellas. El valor puede ser de cualquier tipo. Ejemplo:

```
{ "departamento":8, "nombredpto": "Ventas", "director": "juan rodriguez", "empleados": [{"nombre": "Pedro", "apellido": "Fernandez"}, {"nombre": "Jacinto", "apellido": "Benavente"} ] }
```



Node-Red (Modbus TCP)

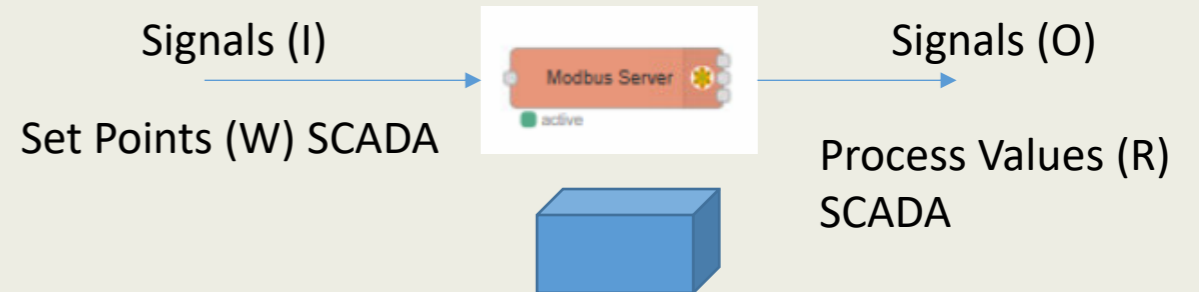


Data exchange via Modbus TCP

SCADA control of multiple distributed IOTs.

SCADA programmed with Visual C# .NET.

Modbus Server Node





SCADA (Modbus TCP)



Control

Comunicación OK
Habilitado

ID: 18
Descripción: 253
Zona: FILA2
IP: 192.168.10.110

Manual (0) / Auto (1)

Controlador Temperatura

PV: 18.1
Set Point: 17.5

Limite Superior SP (°C): 1
Calib. sonda (°C): -1.7

D. Inicial: 1081 g/L D. Actual: 1075 g/L
CHARD ECO CP
CHCP190819

DEN_MAX	DEN_MIN	SP
1083	1059	17.5
1058	1016	14.5
1015	1005	16.5
1004	992	17.5

MENSAJE	DESCENSO_DENSIDAD
OXIGEN 5 mg	20
NUTRICIO + ...	22
CLARIFICACIÓ	30

FECHA DEN DIF TEMP

21/08/2019 9:17	1081		
22/08/2019 8:30	1079		14.0
23/08/2019 8:25	1077	2	13.8
24/08/2019 9:42	1075	2	15.6

Alarmas = 2

18/09/2019 23:46:12

Comunicaciones

250	17,9 °C	SP: 15 °C	D: 1087 g/L	FRESC 1086 (12)	XAR GRAU
251	17,9 °C	SP: 15 °C	D: 1092 g/L	FRESC 1086 (12)	XAR GRAU
252	17,9 °C	SP: 15 °C	D: 1029 g/L	FRESC 1086 (12)	XAR GRAU
253	16,9 °C	SP: 14 °C	D: 1045 g/L	FERM 1073	MCB 1
254	15,5 °C	SP: 14 °C	D: 1024 g/L	FERM 1073	XAR 1
255	15,8 °C	SP: 17,5 °C	D: 1086 g/L	FRESC 1086 (12)	XAR GRAU
256	18,8 °C	SP: 17,5 °C	D: 1083 g/L	FRESC 1086 (12)	XAR GRAU
257	16,8 °C	SP: 17,5 °C	D: 1080 g/L	FRESC 1073 (10)	MCB 1
258	18 °C	SP: 15 °C	D: 1022 g/L	FRESC 1073 (10)	XAR 1
259	16,8 °C	SP: 14 °C	D: 1045 g/L	FERM 1073	XAR 1
260	15,9 °C	SP: 14 °C	D: 1024 g/L	FERM 1079	MAC 1
261	15,5 °C	SP: 17,5 °C	D: 1086 g/L	FRESC 1086 (12)	XAR GRAU
262	22 °C	SP: 20 °C	D: 8 g/L	DESACTIVADO	DESACTIVADO
263	19,5 °C	SP: 18 °C	D: 9 g/L	FERM 1066 (B)	MACABEU ECO
264	16,4 °C	SP: 18 °C	D: 1001 g/L	FERM 1066 (B)	XAR MAC
265	15,7 °C	SP: 14 °C	D: 1085 g/L	FERM 1066 (B)	XAR MAC
266	19,4 °C	SP: 18 °C	D: 9 g/L	FERM 1079	XAR 2ES
267	14,6 °C	SP: 14 °C	D: 1020 g/L	FERM 1073	MAC ECO
268	24,2 °C	SP: 14,3 °C	D: 8 g/L	DESACTIVADO	DESACTIVADO
269	16,6 °C	SP: 20 °C	D: 8 g/L	DESACTIVADO	DESACTIVADO
270	11,6 °C	SP: 18 °C	D: 990 g/L	CHARD TRANQUIL	CHARD TRANQUIL
271	14,8 °C	SP: 14 °C	D: 1044 g/L	FERM 1079	XAR MAC
272	14,6 °C	SP: 14 °C	D: 1017 g/L	FERM 1086 (12)	MAC GRAU
273	14,6 °C	SP: 14 °C	D: 1040 g/L	FERM 1079	MCB 1

FECHA_INICIO DEPOSITO TEMPERATURA ESTADO_ALARMA MENSAJE

18/09/2019 23:45:41	267	20.4	UNACK_ALM	TEMPERATURA FUERA DE MARGEN LO/HI
18/09/2019 18:14:39	248		UNACK_ALM	TEMPERATURA FUERA DE MARGEN LO/HI



MQTT



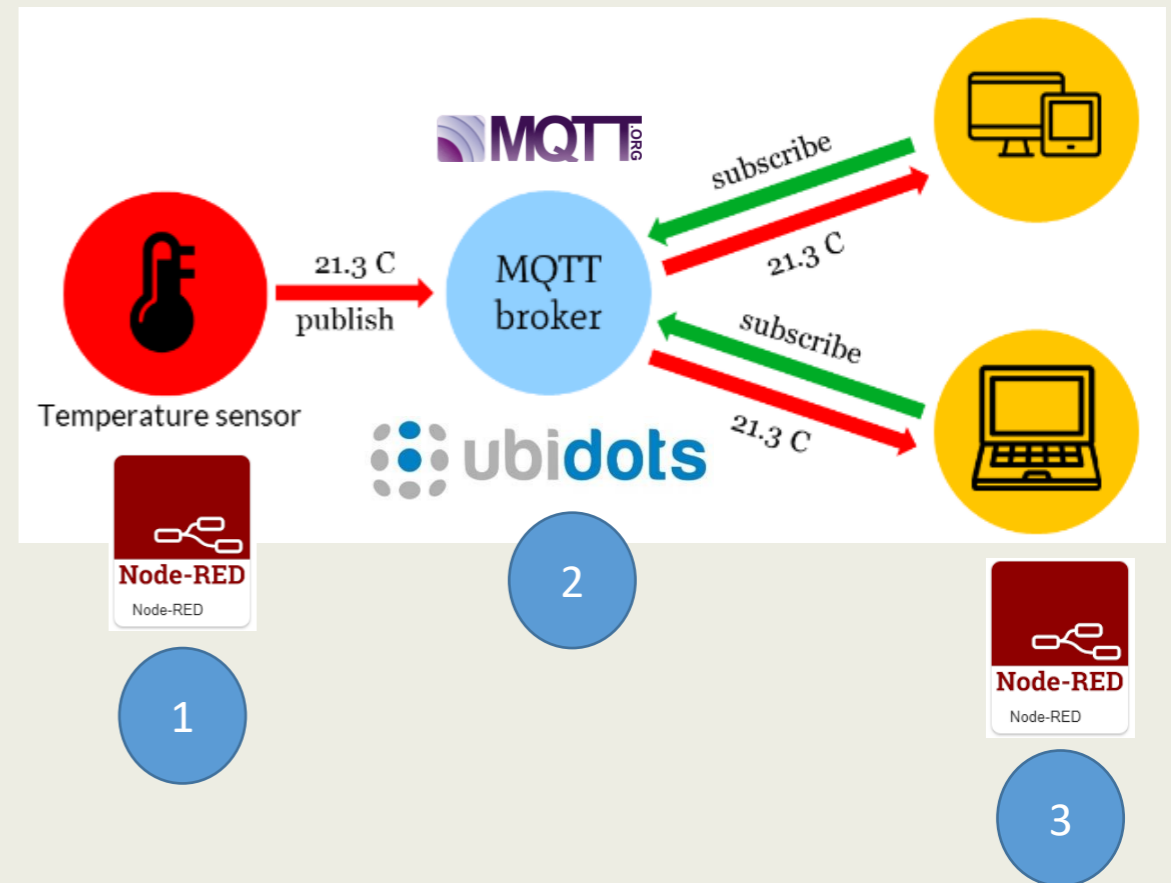
Message Queue Telemetry Transport)

ISO standard publish-subscribe messaging protocol.

Works over TCP / IP.

It is designed for connections to remote locations with "small code footprint" or limited network bandwidth

Broker. Data collector from **publisher** and server to **subscribers**

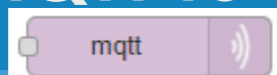


Cloud data Pub/Sub



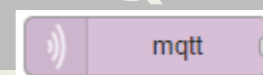
1

Publish MQTT to NodeRed



3

Subscribe MQTT to NodeRed



SUBSCRIPTION
Device: plc_1200
Variable: brightness

Account created
in Ubidots
(broker)

PUBLISH
Device: plc_1200
Variable: msg.payload.brightness

Edit mqtt out node

Delete Cancel Done

Properties

Server: Ubidots

Topic: /v1.6/devices/plc_1200

QoS: 0 Retain:

Name: Name

Edit mqtt out node > Edit mqtt-broker node

Delete Cancel Update

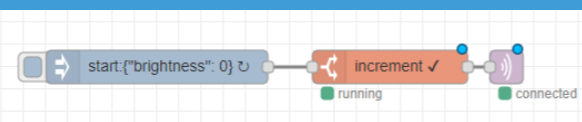
Properties

Name: Ubidots

Connection: Security Messages

Username: BBFF-cEuGo50WfUjNZYokdMGPkKW1SnDH1jBb

Password: [redacted]



Edit mqtt out node > Edit mqtt-broker node

Delete Cancel Update

Properties

Name: Ubidots

Connection: Security Messages

Server: industrial.api.ubidots.com Port: 8883

Enable secure (SSL/TLS) connection

TLS Configuration: TLS configuration

Client ID: Leave blank for auto generated

Keep alive time (s): 60 Use clean session

Use legacy MQTT 3.1 support

Edit mqtt in node

Delete Cancel Done

Properties

Server: Ubidots

Topic: /v1.6/devices/plc_1200/brightness

QoS: 2

Output: auto-detect (string or buffer)

Name: Name

mqtt /v1.6/devices/plc_1200/brightness connected

msg.payload

debug

```
28/8/2019 16:32:24 node: 301bc5df.14d6ea
running : msg.payload.brightness : number
30
28/8/2019 16:32:25 node: 301bc5df.14d6ea
running : msg.payload.brightness : number
40
28/8/2019 16:32:26 node: 301bc5df.14d6ea
running : msg.payload.brightness : number
50
28/8/2019 16:32:27 node: 301bc5df.14d6ea
running : msg.payload.brightness : number
60
```




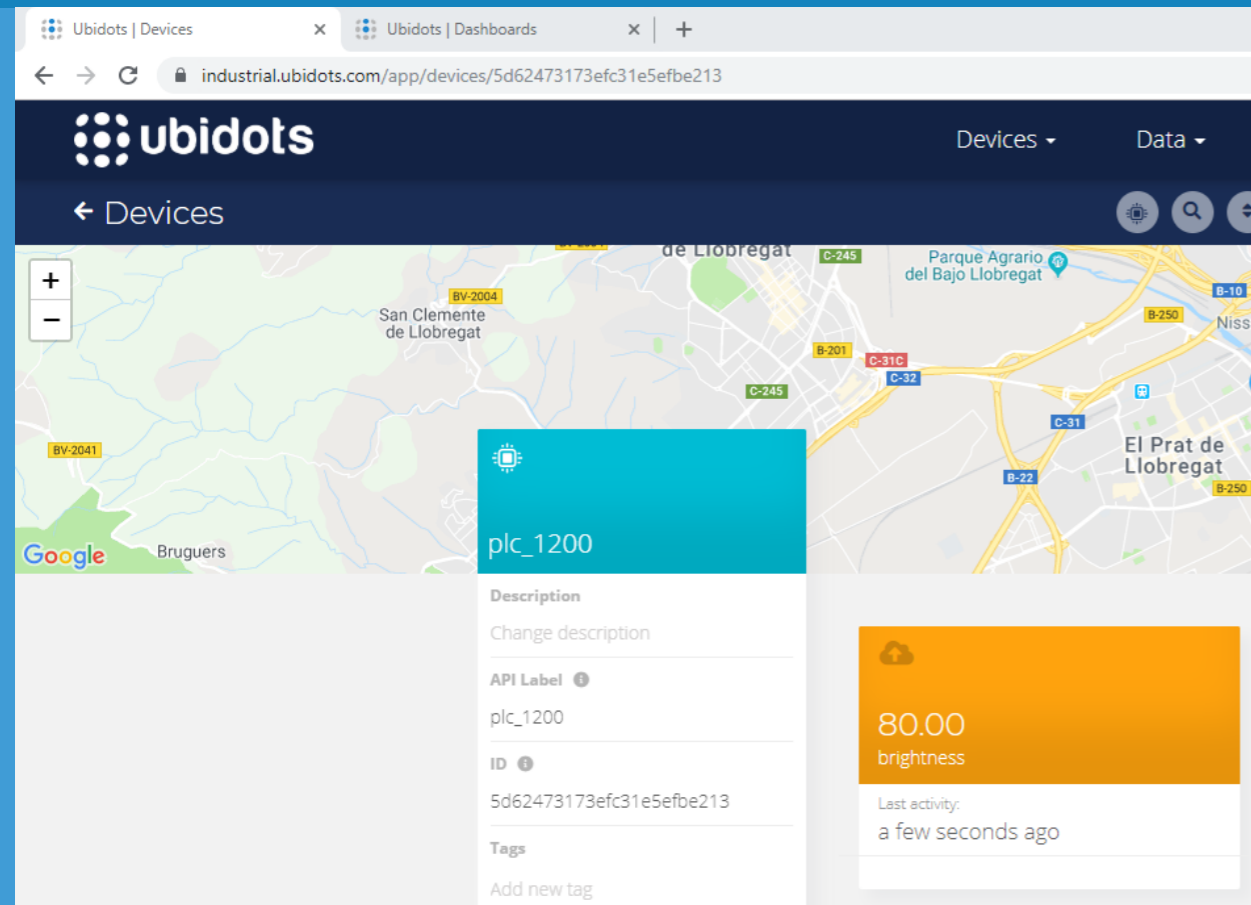
Cloud data broker



2 Ubidots (MQTT broker)

Publishing dots to the cloud from Node-RED

We use MQTT Protocol (Message Queue Telemetry Transport) 



Ubidots | Devices

industrial.ubidots.com/app/devices/5d62473173efc31e5efbe213

ubidots

Devices Data

← Devices

Map showing location: de Llobregat, San Clemente de Llobregat, Bruguers, El Prat de Llobregat.

plc_1200

Description: Change description

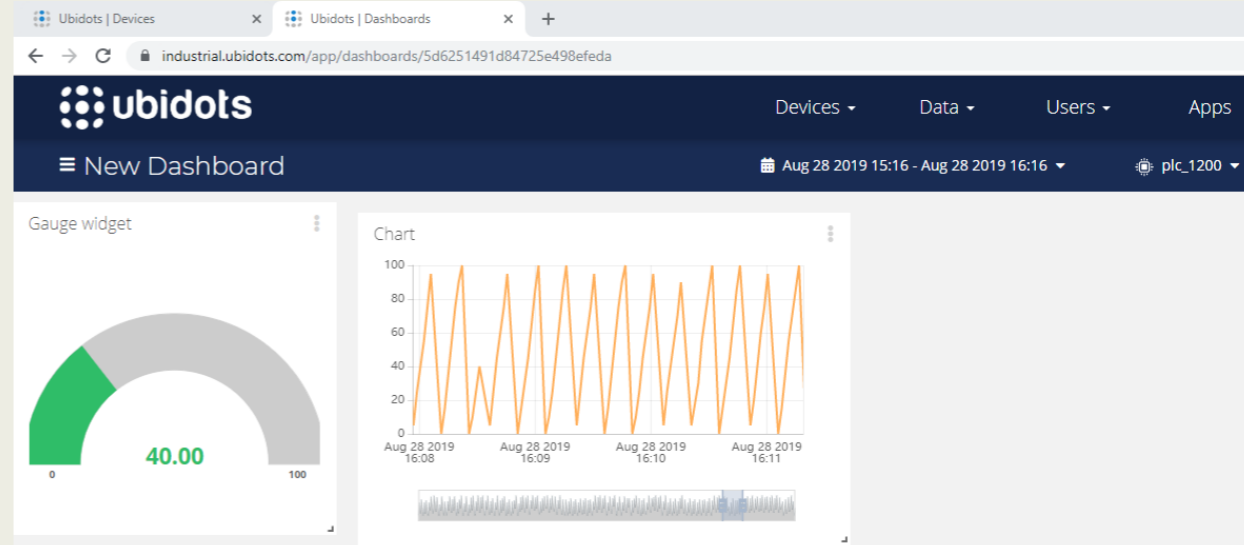
API Label: plc_1200

ID: 5d62473173efc31e5efbe213

Tags: Add new tag

80.00 brightness

Last activity: a few seconds ago



Ubidots | Devices

industrial.ubidots.com/app/dashboards/5d6251491d84725e498efeda

ubidots

Devices Data Users Apps

New Dashboard

Aug 28 2019 15:16 - Aug 28 2019 16:16

plc_1200

Gauge widget: 40.00

Chart: Aug 28 2019 16:08 - Aug 28 2019 16:11

Internet of Things

v.1.2 SEPTEMBER 2024



<https://www.linkedin.com/in/ricardo-moraleda-gareta-9421099>

<https://www.linkedin.com/company/gdo-electric1996/>

<https://www.linkedin.com/in/melvin-francis/>