U.S. DEPARTMENT OF ENERGY'S
CYBERFORCE® PROGRAM

CyberForce® 101

ICS & Node-RED

October 2023

cyberforcecompetition@anl.gov

ICS and Node-RED Overview

Overview

Industrial control systems (ICS) is a collective term used to describe different types of control systems with their associated instrumentation, including the devices, systems, networks, and controls used to operator and/or automate industrial processes. The devices and protocols used in ICS are used in most critical infrastructure sectors like manufacturing, transportation, and energy.

Types of ICS Systems

Supervisory Control and Data Acquisition (SCADA)

SCADA systems provide control at the supervisory level and are made up of components at different places, including Programmable Logic Controllers (PLC) and other hardware modules. They can acquire and transmit data and are integrated with a Human Machine Interface (HMI) that provides central monitoring and control for process inputs and outputs.

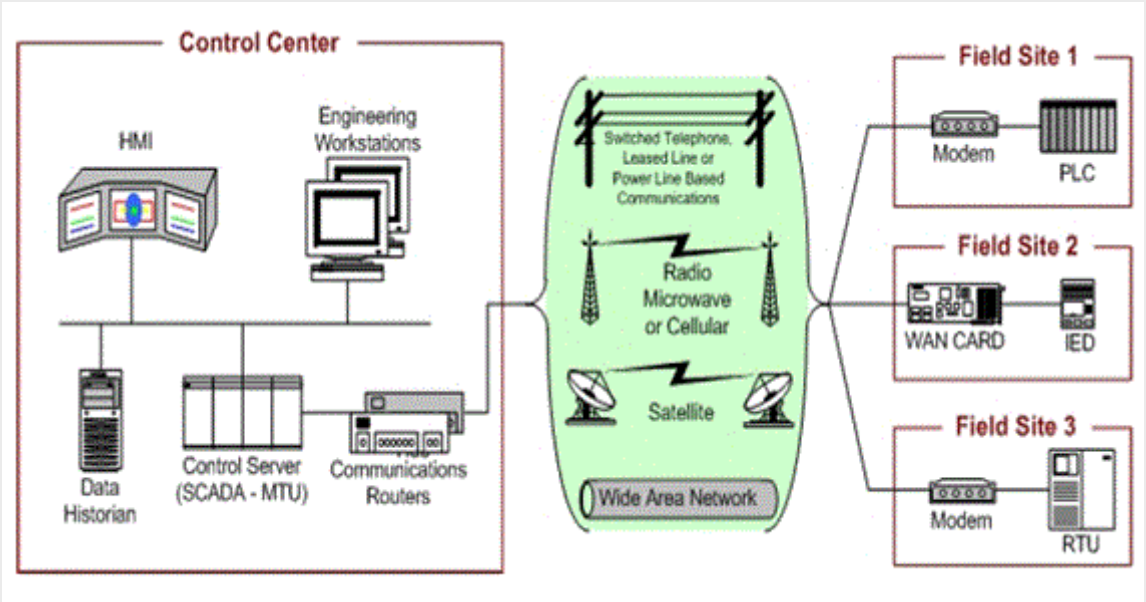
It's mostly used for remote monitoring and management of field sites. It allows for automation so that employees do not have to physically complete tasks or collect data. Local field devices control valves and breakers, sensor systems, and environment monitoring for alert situations.

Distributed Control System (DCS)

DCS systems control production systems found in one location. A setpoint communicates to the controller, controlling how valves or actuators operate to maintain the intended setpoint. Field data is used for process control or for future plans.

Each DCS utilizes a centralized supervisory control loop to manage multiple controllers and devices. This way we can access production and operation data quickly. It also lessens the effect of a single defect on the entire system by using several devices during production.

Components of an ICS Environment



From Survey of ICS Security

IT and OT

Operational technology (OT) includes the hardware and software systems that monitor and control the physical devices in the field. OT tasks change depending on the industry.

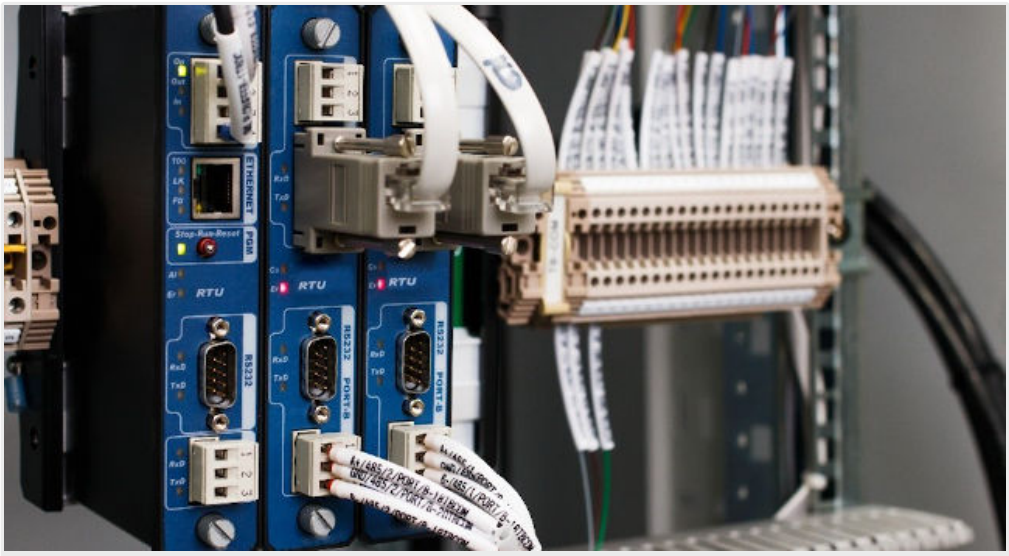
PLC

Programmable Logic Controllers are hardware used in ICS to control the overall system. It provides local management of processes being run through feedback control devices like sensors.

From MISUMI

RTU

Remote Terminal Units are microprocessor-controlled field devices. They receive commands and send information to a SCADA server or master terminal unit.



From Control Engineering

Control Loop

Control loops consist of hardware like PLCs and interprets signals from sensors, control valves, switches, and other devices. The data from these sensors are then sent to the controller that carries out a task or completes a process.

HMI

A Human Machine Interface is a graphic user interface application that allows the human operator to interact with the controller hardware. It can show the status information and historical data from ICS devices. It can be used to monitor and configure setpoints, control algorithms, and adjust parameters in the controllers.

From Copadata

Control Server

The control sever hosts the PLC supervisory control software, as well as communicates with lower-level control devices.

SCADA Server

This device issues commands to RTUs in the field.

IED

An Intelligent Electronic Device acquires data, communicates with other devices, and performs local processing and control. It allows for controls at the local level to be done automatically.

Data Historian

A Data Historian is a centralized database that logs all process information and exports data to the business. The data gathered is used for process analysis, statistical process control, and enterprise planning.

Communication within ICS Systems

Devices and hardware in ICS systems relay important information through communication protocols. There are many different protocols used, and most of these are designed for specific purposes. They were developed to ensure interoperability between different manufacturers.

Modbus

Modbus uses serial communications with the PLCs and is often used in ICS environments. There are two types of Modbus implementations: Serial Modbus and Modbus-TCP. Serial Modbus uses high-level data link control (HDLC) standard for data transmission. Modbus-TCP uses the TCP/IP protocol stack to transmit data.

Node-RED Overview

Node-RED is a programming tool for wiring together hardware devices, APIs, and online services. It is built on Node.js and takes advantage of its event-driven, non-blocking model. You can run it locally, on a device, or in the cloud.

Each node has a purpose. It is given data and does something with it before passing it on. The network is responsible for the flow of data between the nodes.

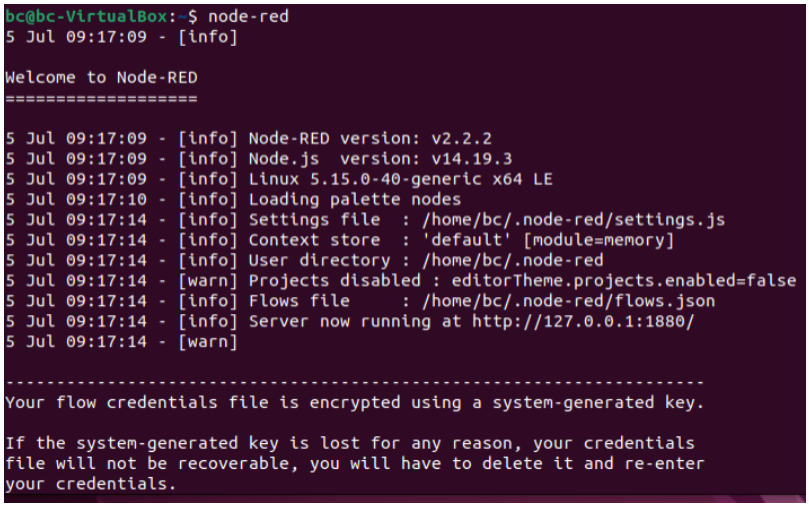
Through a web browser, you can access the flow editor. You can create your application by dragging nodes from the palette into the workspace and then wire them together. You can deploy your application through a single click on the "Deploy" button.

Installation

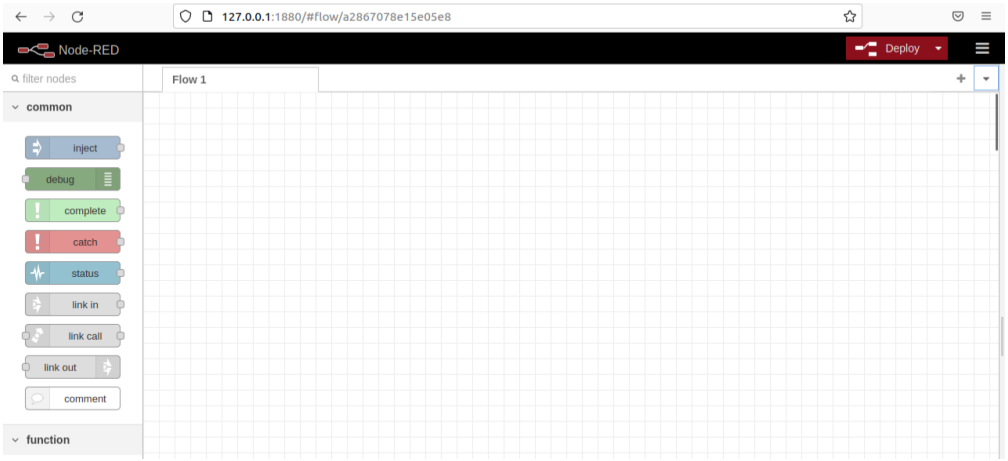
You can use the `npm` command that comes with node.js. If Linux, use `sudo`. If using Windows, do not use `sudo`.

```
sudo npm install -g --unsafe-perm node-red
```

To run Node-RED, use the `node-red` command in your terminal. To quit Node-RED, use `Ctrl-C`.

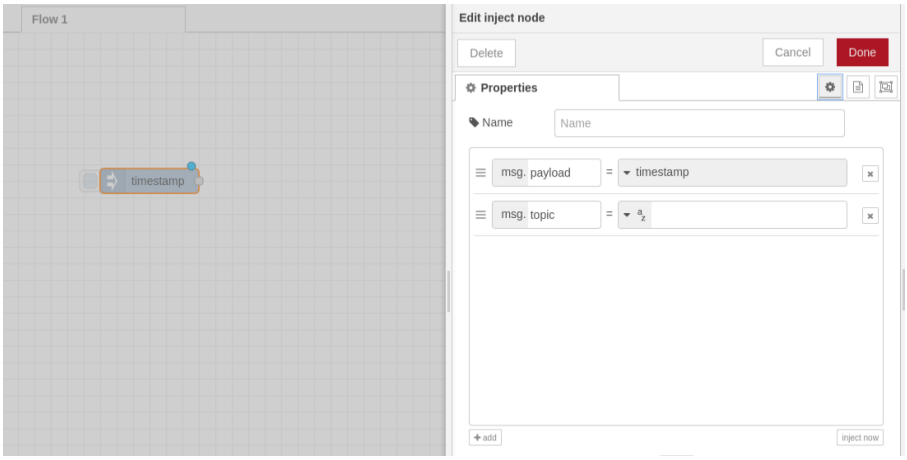


You can access the editor by pointing your browser at http://localhost:1880.



Example: Creating a Flow (add images)

After accessing the editor through the browser, add an Inject node by dragging it onto the workspace from the palette. The Inject node allows you to inject messages into a flow, either through manual clicks or through time intervals between injects. If you select the node by double clicking it, you can see information about its properties and a description of its function in the sidebar pane "Information."

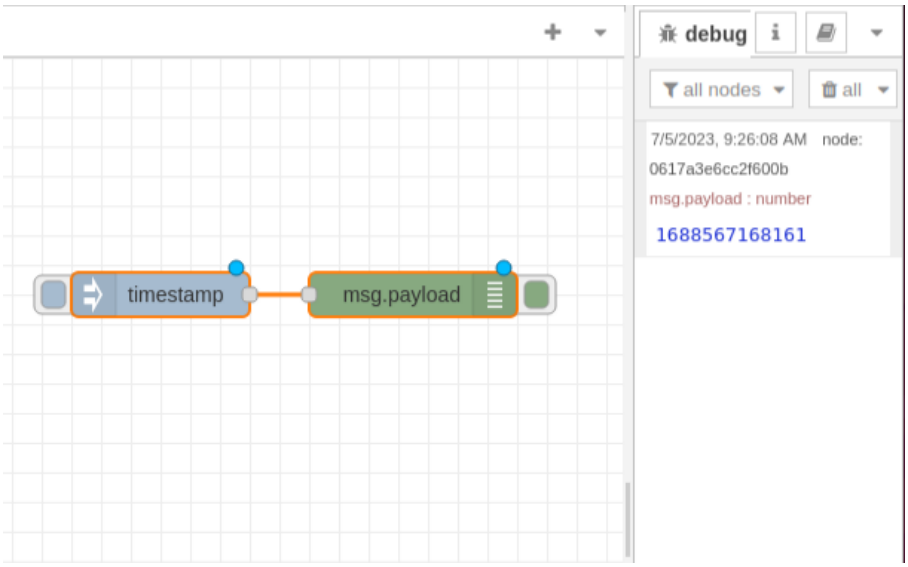


Next, add a Debug node, which will cause any message to be displayed in the Debug sidebar. It will display the payload of the message by default, but you can make it display the entire message object.

Now, connect the Inject and Debug nodes together by dragging the output port of one to the input port of the other.

Then, use the Deploy button to deploy it to the server.

Make sure to select the Debug sidebar. Click the Inject button (the small square button next to the node) and numbers should appear in the sidebar.

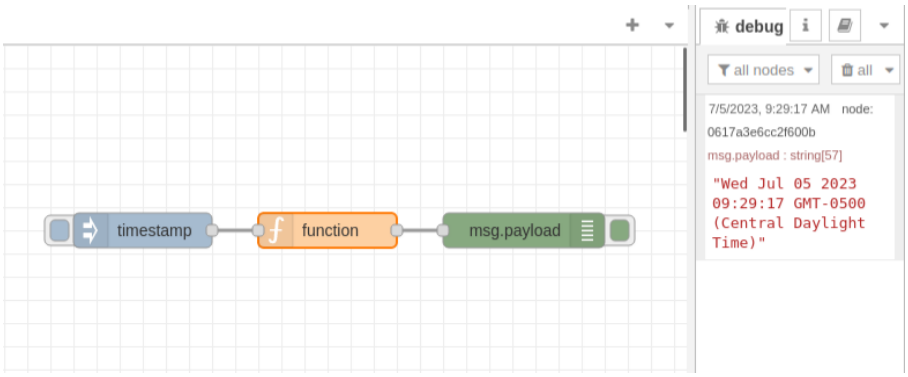


Lastly, add a Function node. This allows you to pass each message through a JavaScript function. Delete your existing wire by selecting it and pressing delete on the keyboard. Wire the Function node in between the Inject and Debug nodes. Double-click the Function node and edit the dialog with the following code.

```
// Creates a Date object from the payload
var date = new Date(msg.payload);
// Change the payload to a formatted Date string
```

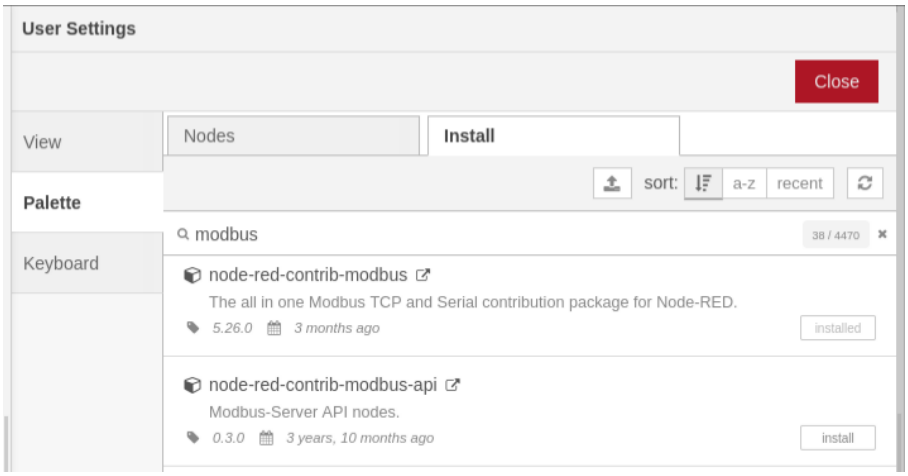
```
msg.payload = date.toString();
// Return the message to send it to the Debug node
return msg;
```

Click Done to close the edit dialog and then click Deploy. Now when you click Inject, the messages in the sidebar will be timestamps.



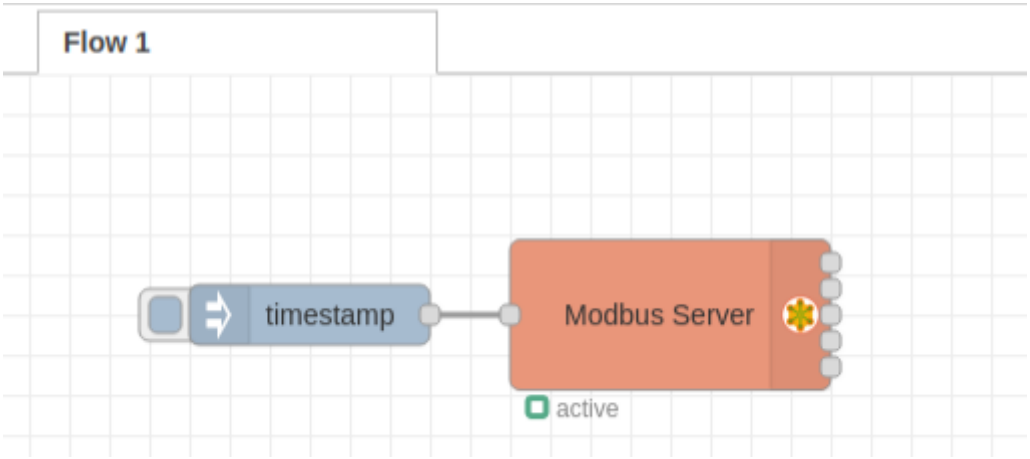
Modbus in Node-RED

To use Modbus in Node-RED, you need to install the nodes for Modbus. By going to the menu and selecting manage Palette, you can search for the `node-red-contrib-modbus` package. There should be 11 nodes in the package.



Modbus Server Configuration

The Modbus getter and write nodes require connection to a server. For a TCP/IP server, you need to configure the IP address and the port number. The default modbus server port is 502.



Node Status

The getter and write nodes display the server status under the node. A green active status shows that we are connected to the Modbus server. The status will change as we read or write data.

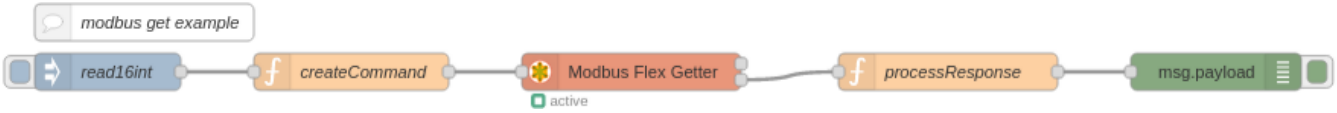
Read Nodes

The modbus getter and modbus flex getter can be used for reading data using Modbus TCP/IP, TCP RTU, and serial. The read mode is configured in the server properties.

The modbus getter node already has the configuration done in the node. You use this when doing a fixed read.

The modbus flex getter must have the configuration data passed in from a preceding node. You should use this when you are doing multiple reads of different values.

In the example below, we use the inject node to trigger the flow. The function node creates the read command and defines the configuration for the flex getter node. The processResponse function node extracts the data from the modbus server before its output is sent to the debug node.



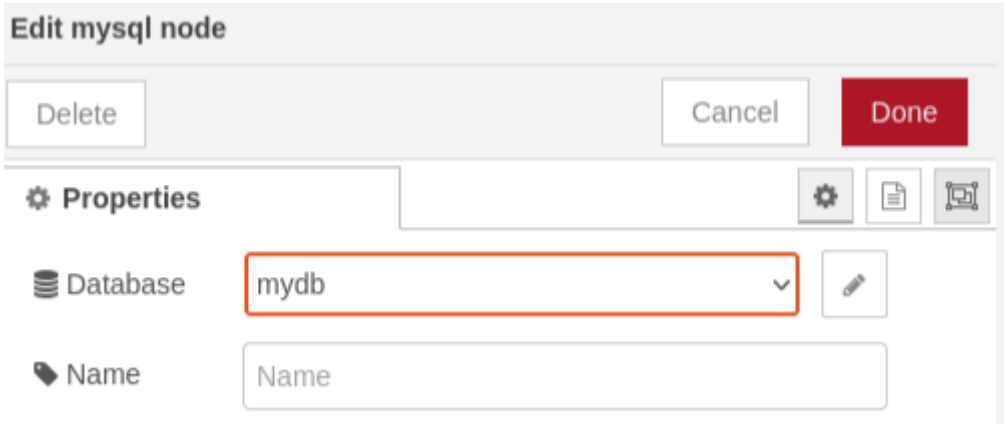
Write Nodes

The modbus write and modbus flex write nodes are used for writing data to a modbus server. The write node has all of the configuration done in the node itself whereas the flex write gets the configuration from the preceding node.

In the example below, the function node is used to create the write command and its contents get sent to the flex write node which writes the values to the modbus server.

MySQL in Node-RED

To get MySQL nodes in Node-RED, you can install the package from the palette like we did for the Modbus nodes. These nodes will allow us to have basic access to a MySQL database. To first use it, you must edit the node to connect it to a database.

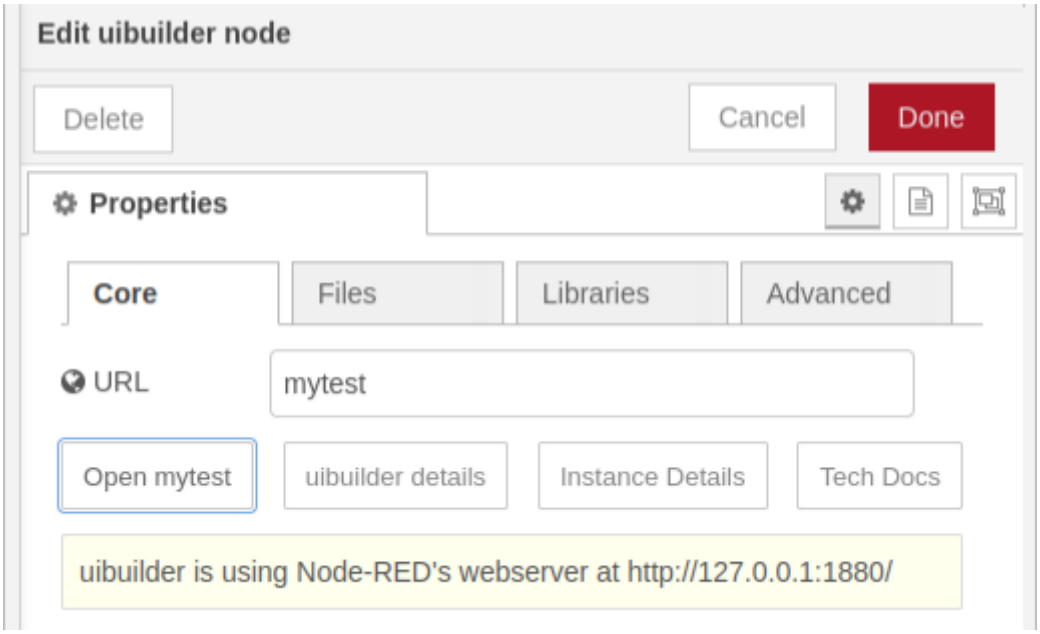


It uses the query operation against the configured database, allowing us to use `INSERT`'s and `DELETE`'s. The `msg.topic` must hold the query for the database, and the result will be returned in `msg.payload`.

uiBuilder in Node-RED

Using uibuilder, you can create a stand-alone web server that works with Node-RED. You can install it through the palette manager.

To use it, first add a uibuilder node. Open the settings and give it a "URL" for the identifying name. Then you can deploy it. If you add an inject node for input data and two debug nodes on the two output ports, you can see what's going on after deploying it again. Now you can click on the uibuilder node and click "Open" to see the web page.



Exercise

These three types of nodes are found heavily in CyberForce competitions. To practice, we have created an exercise that combines all of these nodes together.

You will set up a Modbus server first. Then you will write data to a modbus server before getting it from the modbus server and creating a query for its values to be inserted into a SQL database. The data will then be pulled from the SQL database to be displayed on a website using the uibuilder node.

Sources

1. Industrial Control System
2. Industrial Control System (ICS): Definition, Types, Security
3. Node-RED Home Page
4. How to Use Node-Red with Modbus
5. node-red-node-mysql
6. node-red-contrib-uibuilder
7. Writing Modbus Data with node-red
8. Survey of Industrial Control Systems Security
9. MISUMI Mech Lab Blog
10. RTU and SCADA systems help with telemetry monitoring, control
11. What is HMI?