

Tutorial SVG Node (SCADA-Like) Machine Monitoring System Node-Red

Yaser Ali Husen

Overview

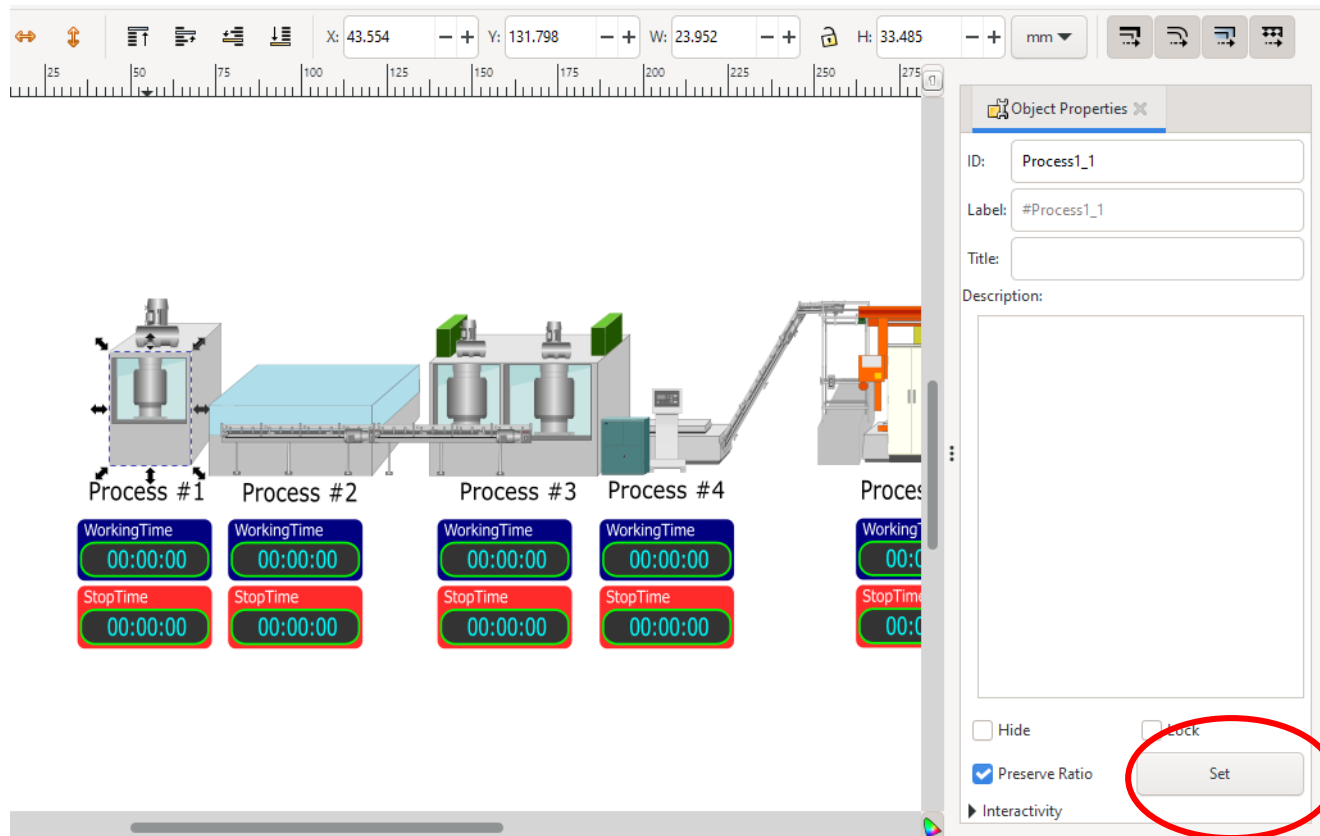
This tutorial contains How to develop a SCADA-Like Dashboard using Node-Red
We can develop a machine monitoring dashboard like SCADA using Node-Red.

Additional Software:

1. Inkscape
2. Symbol Factory (optional)

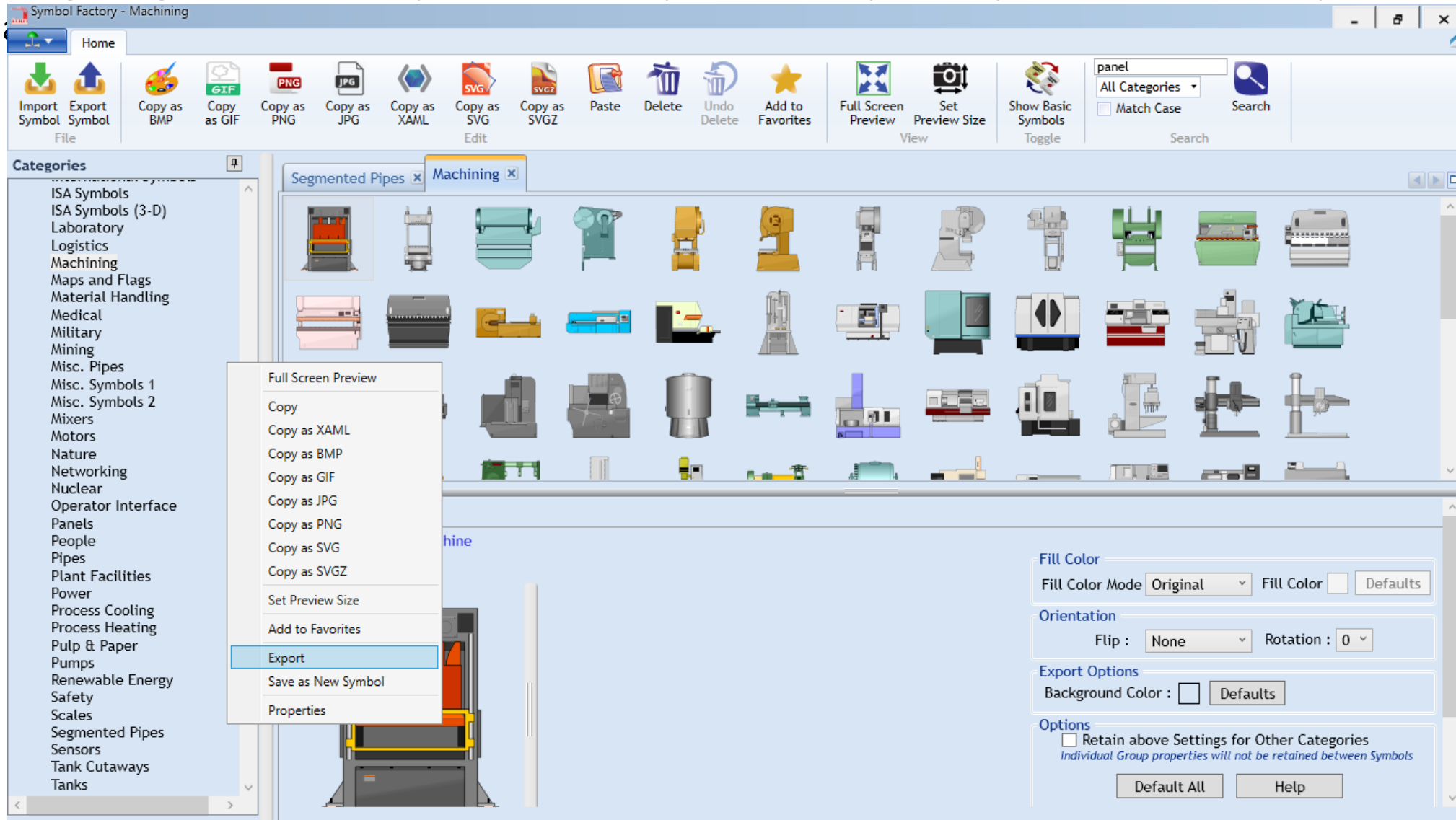
Make SVG File

1. Download and install inkscape
Link: <https://inkscape.org/>
2. Open inkscape, make SCADA dashboard, and save as SVG file
3. For importing a SVG file, klik File → Import...
4. Set ID for each object that want to modify in Node-Red
Right click an object, click Object Properties..., fill ID name and click Set button



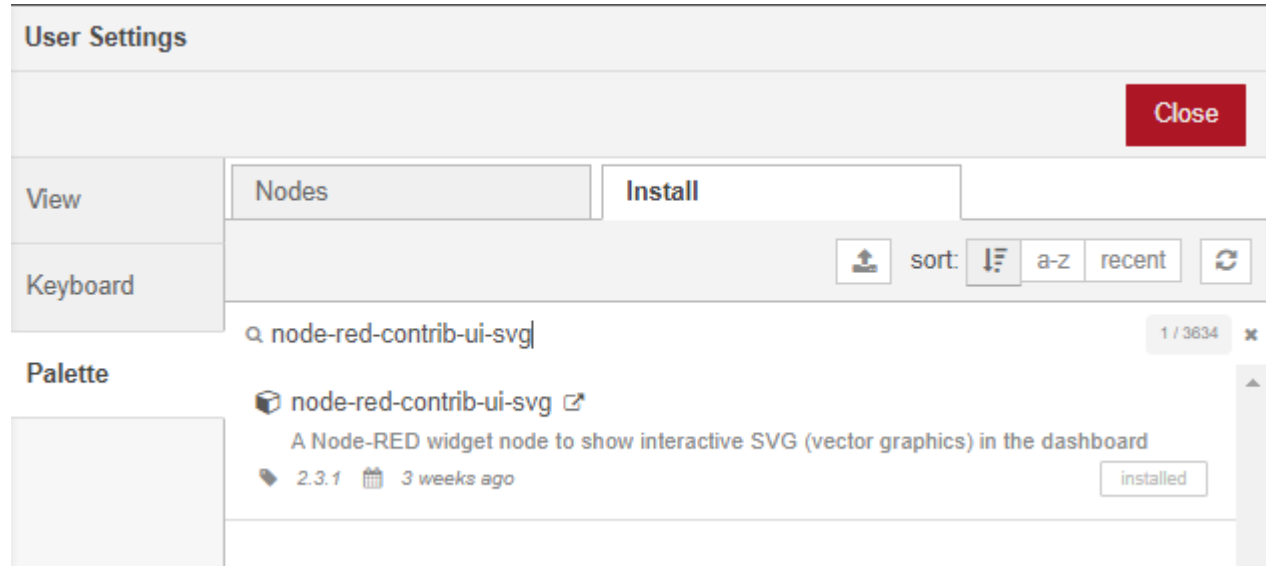
Make SVG File

5. For getting some SVG file, you can install Symbol Factory and export to a SVG file, import in inkscape

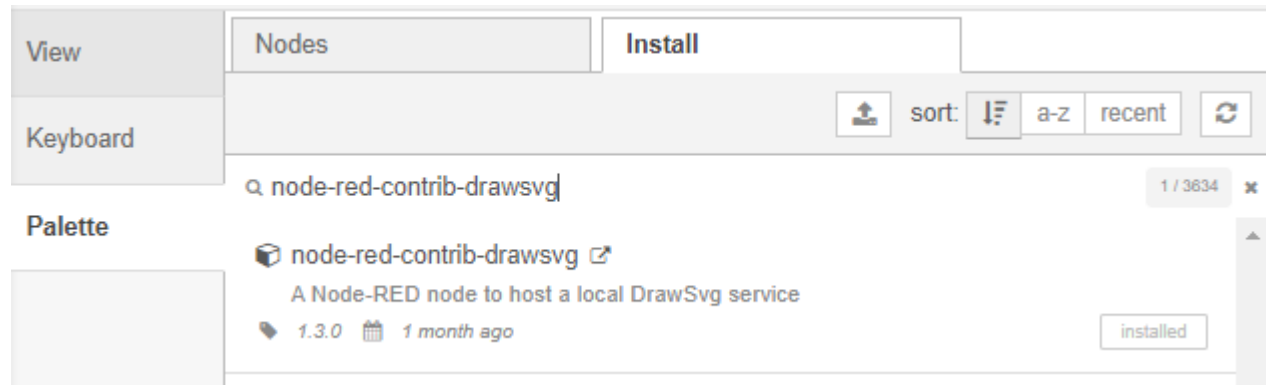


Node-Red Preparation

1. Install node-red-contrib-ui-svg

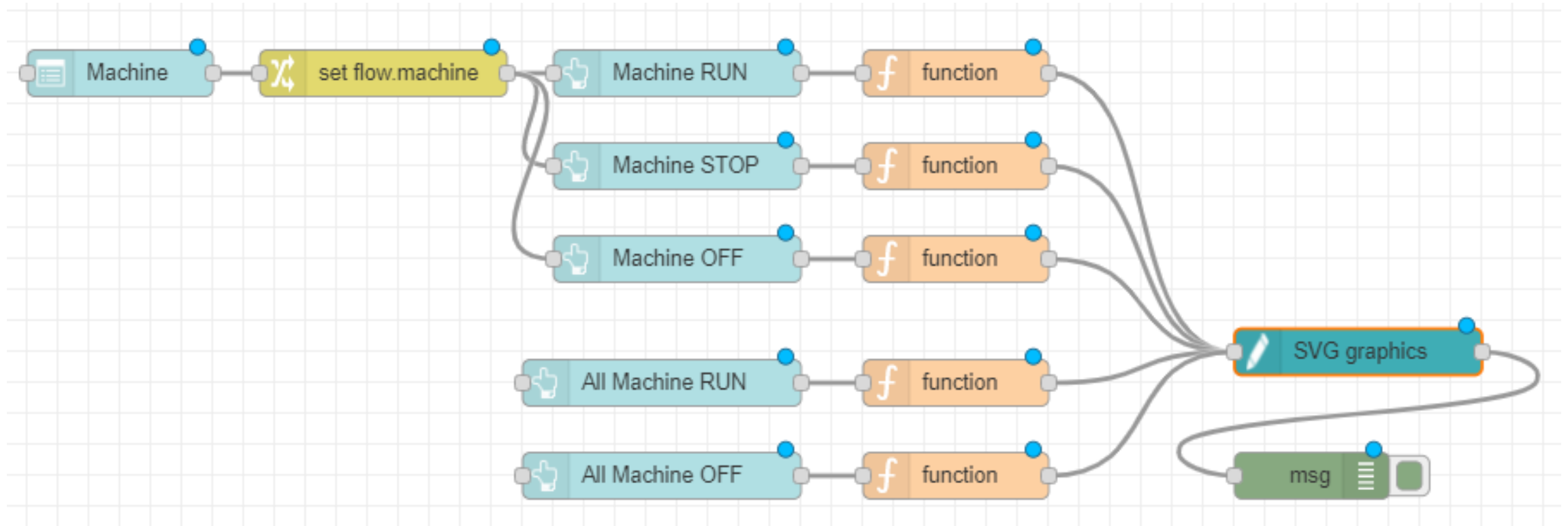


2. Install node-red-contrib-drawsvg



Make Node-Red Dashboard

1. Import sample flow



Make Node-Red Dashboard

2. Setup SVG Editor location, click node SVG graphics, open tab Settings, fill Editor URL

Editor	SVG	Animate	Event	JS	Binding	Settings	CSS
--------	-----	---------	-------	----	---------	----------	-----

☐ Show mouse coordinates (as tooltip)

☒ Auto format SVG Source after saving edits in SVG Editor

☐ Show browser errors on the server

☐ Show browser events on the server

☐ Enable JS event debugging

☐ Send output msg when the client is (re)loaded

☐ Don't send click events in case of a double click event

🌐 Editor URL

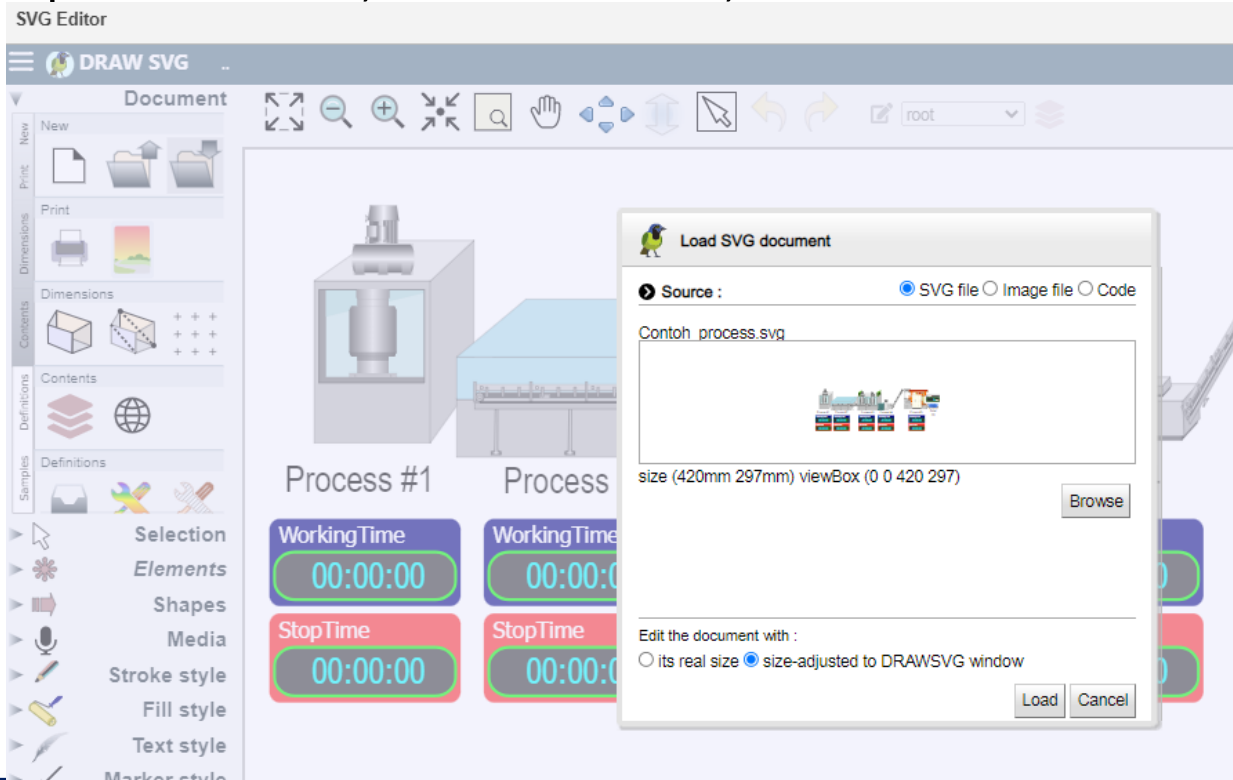
📁 Directory

Make Node-Red Dashboard

3. Load SVG file, click tab Editor, click icon



4. Open document, browse SVG file, click load



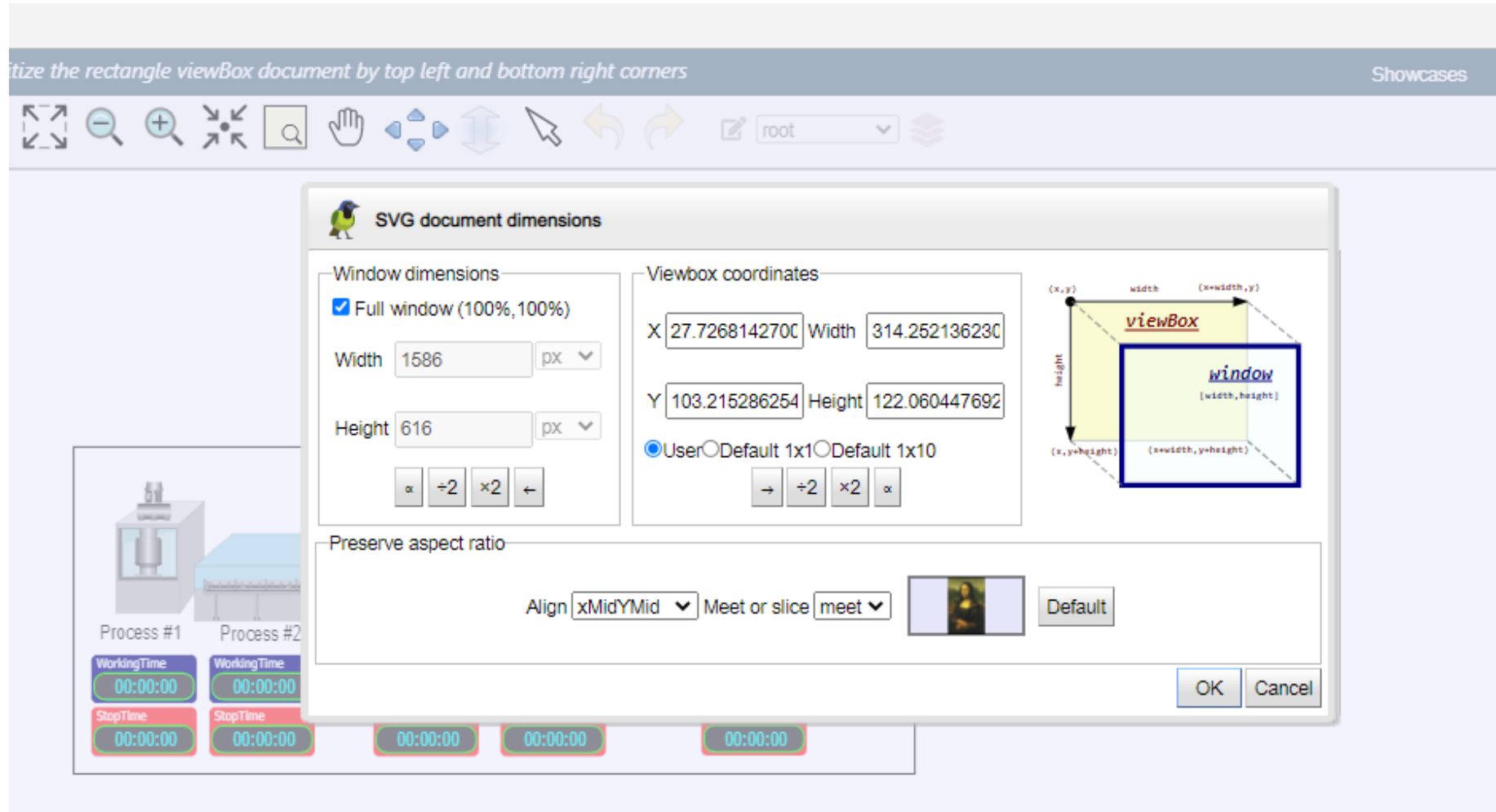
- ☐ Document → Open
- ☐ Browse SVG file location
- ☐ Click Load button

Make Node-Red Dashboard

5. Resize Dimension, Click Resize icon



6. Choose Dimension Area, Check Full window, and OK



Make Node-Red Dashboard

7. Check and set ID object (For text object, set ID with `_value`, example: `output_value`)

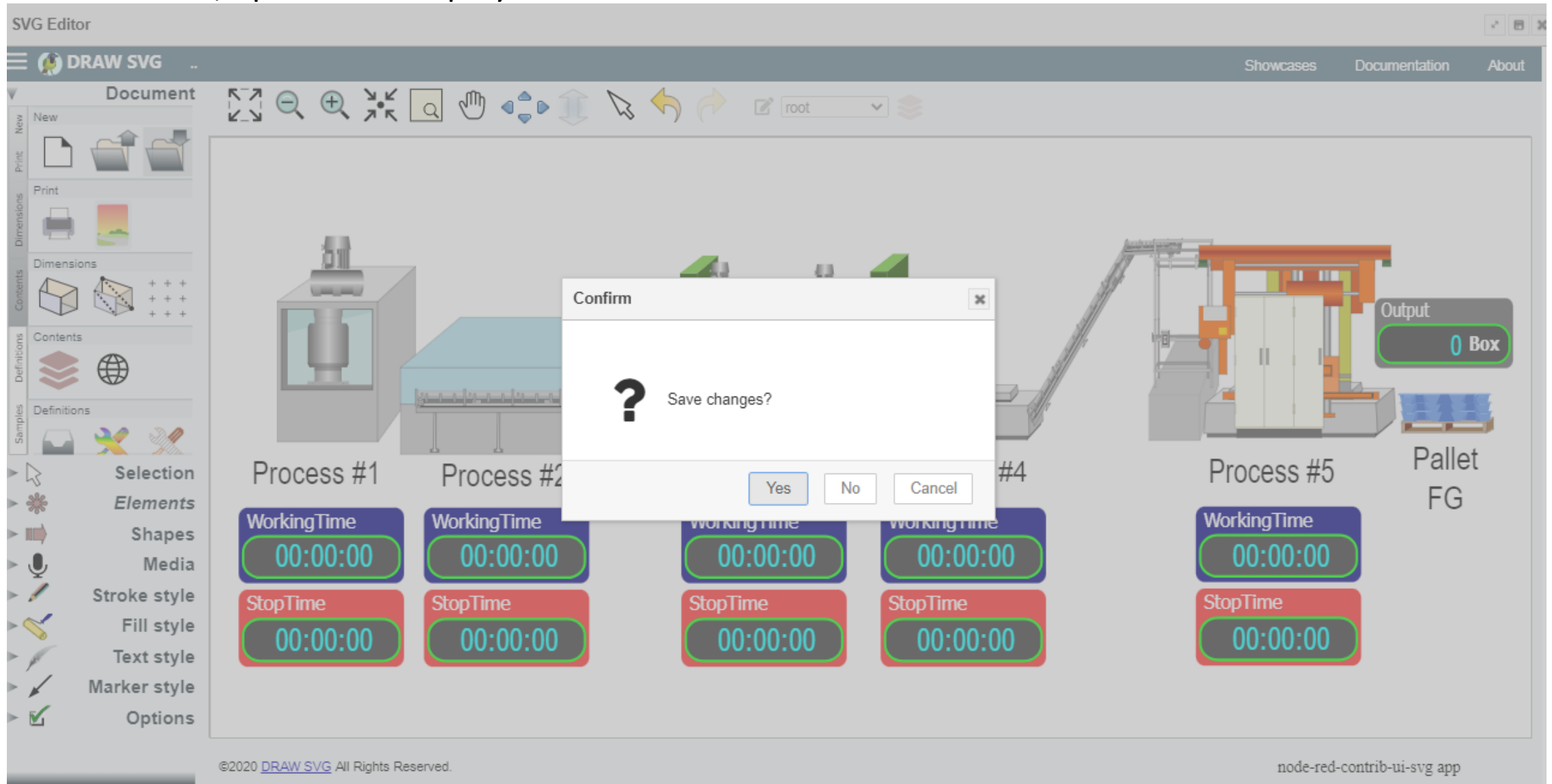
The screenshot shows the Node-RED web interface in a browser window. The address bar displays `localhost:1880/#flow/1190a0ff.bb141f`. The main workspace contains an SVG diagram of a manufacturing process with two stations, 'Process #1' and 'Process #5', each featuring 'WorkingTime' and 'StopTime' digital displays. An 'Output' box shows '0 Box' and a 'Pallet FG' is also visible. An 'SVG Editor' window is open over the workspace, displaying a 'DRAW SVG' toolbar on the left and a 'text properties' panel in the center. The 'text properties' panel has tabs for 'Element', 'Stroke style', 'Fill style', and 'Text style'. The 'Element' tab is active, showing the following details:

Property	Value
Tag	text
Xpath	id('layer1-0')/svg:text[1]
Default style	TEXT
ID	output_value
Transform	scale(0.95087635, 1.05166)
Layer	layer1-0

At the bottom of the SVG Editor window, there is a copyright notice: '©2020 DRAW SVG All Rights Reserved.' and the text 'node-red-contrib-ui-svg app'.

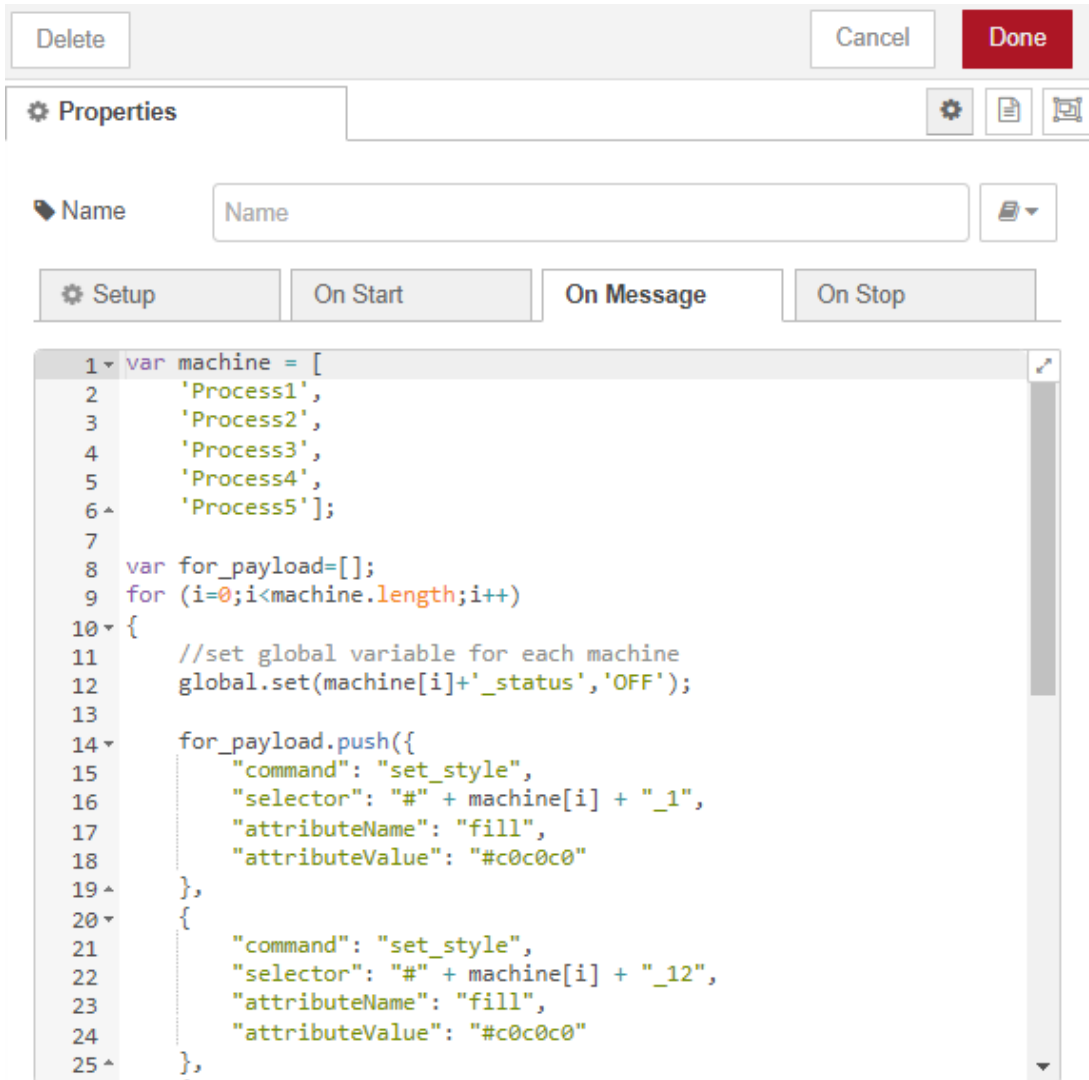
Make Node-Red Dashboard

8. Close and save, update and deploy



Make Node-Red Dashboard

9. Example send command to SVG Change color



The screenshot shows the Node-RED editor interface. At the top, there are buttons for 'Delete', 'Cancel', and 'Done'. Below these is a 'Properties' tab with a 'Name' field. The main configuration area has tabs for 'Setup', 'On Start', 'On Message', and 'On Stop'. The 'On Message' tab is selected, and it contains a JavaScript node with the following code:

```
1 var machine = [  
2   'Process1',  
3   'Process2',  
4   'Process3',  
5   'Process4',  
6   'Process5'];  
7  
8 var for_payload=[];  
9 for (i=0;i<machine.length;i++)  
10 {  
11   //set global variable for each machine  
12   global.set(machine[i]+'_status','OFF');  
13  
14   for_payload.push({  
15     "command": "set_style",  
16     "selector": "#" + machine[i] + "_1",  
17     "attributeName": "fill",  
18     "attributeValue": "#c0c0c0"  
19   },  
20   {  
21     "command": "set_style",  
22     "selector": "#" + machine[i] + "_12",  
23     "attributeName": "fill",  
24     "attributeValue": "#c0c0c0"  
25   },  
26 }
```

Change text



The screenshot shows the Node-RED editor interface. At the top, there are buttons for 'Delete', 'Cancel', and 'Done'. Below these is a 'Properties' tab with a 'Name' field. The main configuration area has tabs for 'Setup', 'On Start', 'On Message', and 'On Stop'. The 'On Message' tab is selected, and it contains a JavaScript node with the following code:

```
1 var time = msg.payload;  
2 msg.payload =  
3 {  
4   "command": "update_text",  
5   "selector": "#wt_Process1_value",  
6   "textContent": time  
7 };  
8 return msg;
```

Make Node-Red Dashboard

10. Open Dashboard

