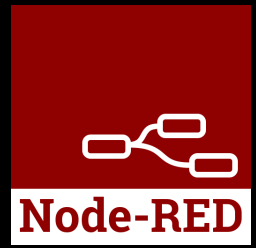
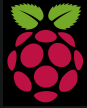


WWW.INFOPULSAS.LT
sales@infopulsas.lt



HOWTO use NodeRED® on C4-8CO Controller



Raspberry Pi is a trademark of the Raspberry Pi Foundation.
More information under www.raspberrypi.org

it's all about perfection _____





STRIVE IN PERFECTION
IN WHATEVER YOU
DO
TAKE THE BEST THAT
EXISTS AND MAKE IT
BETTER
WHEN IT DOES NOT
EXIST. DESIGN IT.

Sir Henry Royce

CONTENT

In this application note you will find ...

PREREQUISITES

Install NodeRED®
on RESI-T4/C4 controller

Our RESI-C4-A-8CO controller

Install NodeRED®
components

First NodeRED®
flow

NodeRED® flow to update
DIP switches+LEDs

NodeRED® flow to update
relay outputs

NodeRED® flow to test
relay outputs

NodeRED® flow to read/write
relay outputs from/to MQTT

NodeRED® flow to build
USER INTERFACE for DIP+LEDs

NodeRED® flow to build
USER INTERFACE for relay outputs

AN-5

AN-7

AN-11

AN-12

AN-13

AN-14

AN-24

AN-27

AN-28

AN-39

AN-46

PREREQUISITES



Raspberry Pi is a trademark of the Raspberry Pi Foundation.
More information under www.raspberrypi.org

PREREQUISITES

We assume that the reader is familiar how to use WINDOWS® operating system, how to configure a LINUX® Ethernet interface, how to use a remote desktop program or SSH console to configure LINUX®. Also we assume that the reader is able to install and open NodeRED® in a browser.

Furthermore we assume, that the reader is able to create a correct NodeRED® flow and that the reader is able to write a JavaScript script. If not, please consult the internet or book a education workshop. RESI is in no way responsible, if you or your customer cannot use the given advice here, because of lack of education in your or their staff!

With the purchase of a IoT Controller from RESI, you have not purchased the right of free education or free consulting from RESI!

RESI delivers IoT controllers with the ability to run NodeRED® on it, but RESI is not liable for any functional problems, software errors, law suits or other issues which results out of using NodeRED® on our devices in your project or machinery!

IMPORTANT SAFETY NOTES

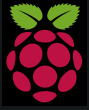
Important hint:

Before you start with the installation and the initial setup of the device, you have to read this document and the attached installation guide and the actual manual for the device very carefully. You have to follow all the herein given information very accurate!

- Only authorized and qualified personnel are allowed to install and setup the device!
- The connection of the device must be done in de-energized state!
- Do not perform any electrical work while the device is connected to power!
- Disable and secure the system against any automatic restart or power on procedure!
- The device must be operated with the defined voltage level!
- Supply voltage jitters must not exceed the technical specifications and tolerances given in the technical manuals for the product. If you do not obey this issue, the proper performance of the device cannot be guaranteed. This can lead to fail functions of the device and in worst case to a complete breakdown of the device!
- You have to obey the current EMC regulations for wiring!
- All signal, control and supply voltage cables must be wired in a way, that no inductive or capacitive interference or any other severe electrical noise disturbance may interfere with the device. Wrong wiring can lead to a malfunction of the device!
- For signal or sensor cables you have to use shielded cables, to avoid damages through induction!
- You have to obey and to apply the current safety regulations given by the ÖVE, VDE, the countries, their control authorities, the TÜV or the local energy supply company!
- Obey country-specific laws and standards!
- The device must be used for the intended purpose of the manufacturer!
- No warranties or liabilities will be accepted for defects and damages resulting from improper or incorrect usage of the device!
- Subsequent damages, which results from faults of this device, are excluded from warranty and liability!
- Only the technical data, wiring diagrams and operation instructions, which are part to the product shipment are valid!
- The information on our homepage, in our datasheets, in our manuals, in our catalogues or published by our partners can deviate from the product documentation and is not necessarily always actual, due to constant improvement of our products for technical progress!
- In case of modification of our devices made by the user, all warranty and liability claims are lost!
- The installation has to fulfill the technical conditions and specifications (e.g. operating temperatures, power supply, ...) given in the devices documentation!
- Operating our device close to equipment, which do not comply with EMC directives, can influence the functionality of our device, leading to malfunction or in worst case to a breakdown of our device!
- Our devices must not be used for monitoring applications, which solely serve the purpose of protecting persons against hazards or injury, or as an emergency stop switch for systems or machinery, or for any other similar safety-relevant purposes!
- Dimensions of the enclosures or enclosures accessories may show slight tolerances on the specifications provided in these instructions!
- Modifications of this documentation is not allowed!
- In case of a complaint, only complete devices returned in original packing will be accepted!



Install NodeRED® on RESI-T4/C4 controller



Raspberry Pi is a trademark of the Raspberry Pi Foundation.
More information under www.raspberrypi.org

it's all about perfection _____

RESI

Install NodeRED® on RESI-T4/C4 controller

Please search in the internet for a tutorial or more information, how to install NodeRED on a Raspberry Pi. We do not want to write yet another manual for the installation of NodeRED.

Open with VNCViewer the Raspberry Desktop or connect your monitor direct to HDMI and keyboard+mouse to the USB interface of our C4/T4 controllers. But do NOT login with root user. Choose your local user like pi. In the desktop choose Settings → Add/remove Software. Enter in the search field Options NodeRED. Select the package and click OK. After a while NodeRED is installed.

We are using NodeRED Version 4.x based on NodeJS version 18 on a 64 Bit OS (bookworm)

WHERE IS NODE RED?

Determine the exact location of the node-red command.

If you have done a global install of node-red, then on Linux/OS X the node-red command will probably be either: `/usr/bin/node-red` or `/usr/local/bin/node-red`. The command which node-red can be used to confirm the location.

If you have done a local install, it will be `node_modules/node-red/bin/node-red`, relative to where you ran `npm install` from.

INSTALL PROCESS MANAGER2

Install pm2 to start/stop the NodeRED system

```
sudo npm install -g pm2
```

HOWTO START/STOP NodeRED

We are using pm2 to manually start or stop NodeRED after system power on.

```
pm2 start /usr/bin/node-red -- -v
```

or

```
pm2 start /usr/local/bin/node-red -- -v
```

```
resi@RESI-C4:~$ pm2 start /usr/local/bin/node-red
[PM2] Starting /usr/local/bin/node-red in fork_mode (1 instance)
[PM2] Done.
```

id	name	mode	u	status	cpu	memory
0	node-red	fork	0	online	0%	36.0mb

You can get info from NodeRED with

```
pm2 info node-red
```

```
pm2 logs node-red
```

You can stop NodeRED with

```
pm2 stop node-red
```



Install NodeRED[®] on RESI-T4/C4 controller

HOWTO AUTOMATICALLY START NODE RED AT SYSTEM STARTUP?

We are using pm2 to automatically start NodeRED after system power on.
First start NodeRED with

```
pm2 start /usr/bin/node-red -- -v  
or  
pm2 start /usr/local/bin/node-red -- -v
```

Then save the current setup with

```
pm2 save  
and  
pm2 startup
```

Then execute the shown command as described:

```
sudo env PATH=$PATH:/usr/bin /usr/local/lib/node_modules/pm2/bin/pm2 startup systemd -u resi --hp /home/resi
```

Finished!



Install NodeRED® on RESI-T4/C4 controller

```
resi@RESI-C4:~$ pm2 startup
[PM2] Init System found: systemd
[PM2] To setup the Startup Script, copy/paste the following command:
sudo env PATH=$PATH:/usr/bin /usr/local/lib/node_modules/pm2/bin/pm2 startup systemd -u resi --hp /home/resi
resi@RESI-C4:~$ sudo env PATH=$PATH:/usr/bin /usr/local/lib/node_modules/pm2/bin/pm2 startup systemd -u
resi --hp /home/resi
[PM2] Init System found: systemd
Platform systemd
Template
[Unit]
Description=PM2 process manager
Documentation=https://pm2.keymetrics.io/
After=network.target

[Service]
Type=forking
User=resi
LimitNOFILE=infinity
LimitNPROC=infinity
LimitCORE=infinity
Environment=PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/local/games:/usr/games
:/usr/bin:/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin
Environment=PM2_HOME=/home/resi/.pm2
PIDFile=/home/resi/.pm2/pm2.pid
Restart=on-failure

ExecStart=/usr/local/lib/node_modules/pm2/bin/pm2 resurrect
ExecReload=/usr/local/lib/node_modules/pm2/bin/pm2 reload all
ExecStop=/usr/local/lib/node_modules/pm2/bin/pm2 kill

[Install]
WantedBy=multi-user.target

Target path
/etc/systemd/system/pm2-resi.service
Command list
[ 'systemctl enable pm2-resi' ]
[PM2] Writing init configuration in /etc/systemd/system/pm2-resi.service
[PM2] Making script booting at startup...
[PM2] [-] Executing: systemctl enable pm2-resi...
Created symlink /etc/systemd/system/multi-user.target.wants/pm2-resi.service → /etc/systemd/system/pm2-resi.service.
[PM2] [v] Command successfully executed.
+-----+
[PM2] Freeze a process list on reboot via:
$ pm2 save

[PM2] Remove init script via:
$ pm2 unstartup systemd
```



NodeRED® sample for RESI-C4-8CO controller

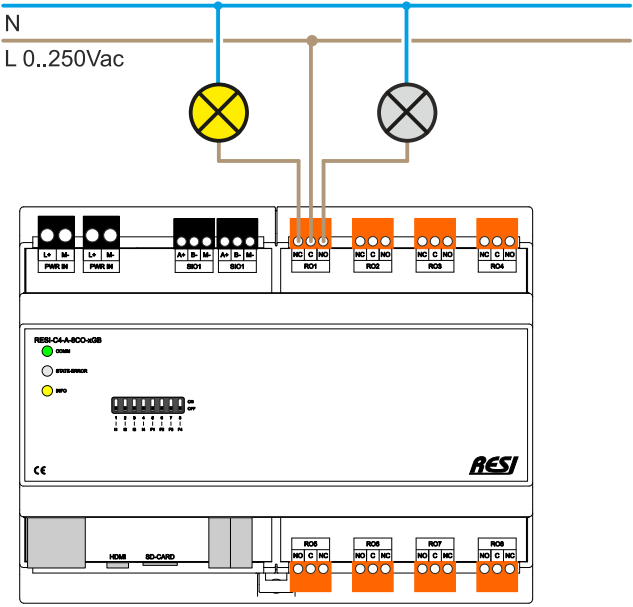


Raspberry Pi is a trademark of the Raspberry Pi Foundation.
More information under www.raspberrypi.org

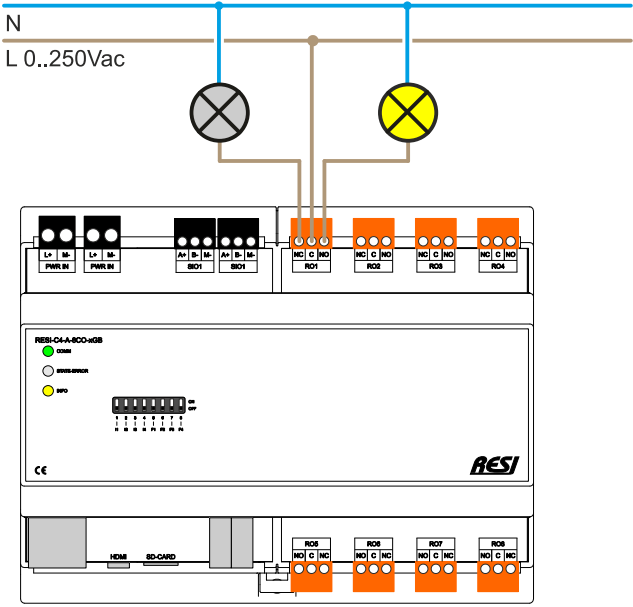
it's all about perfection _____

RESI

NodeRED® on our RESI-C4-A-8CO-xGB controllers



RELAY=OFF



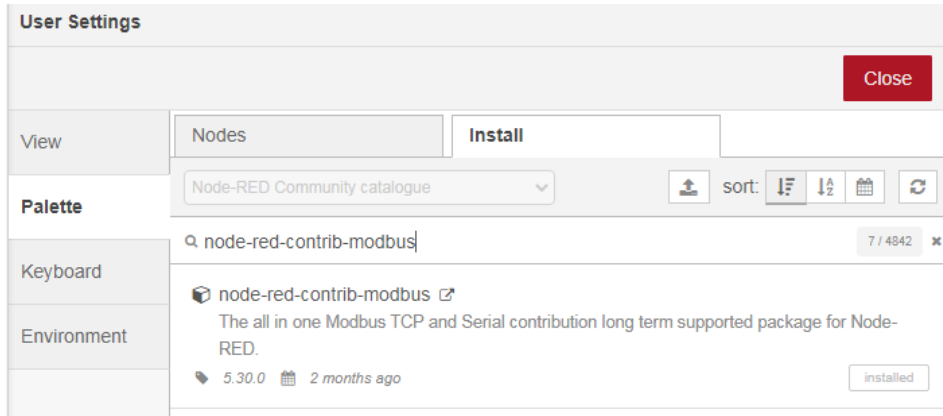
RELAY=ON



Install NodeRED® components node-red-contrib-modbus

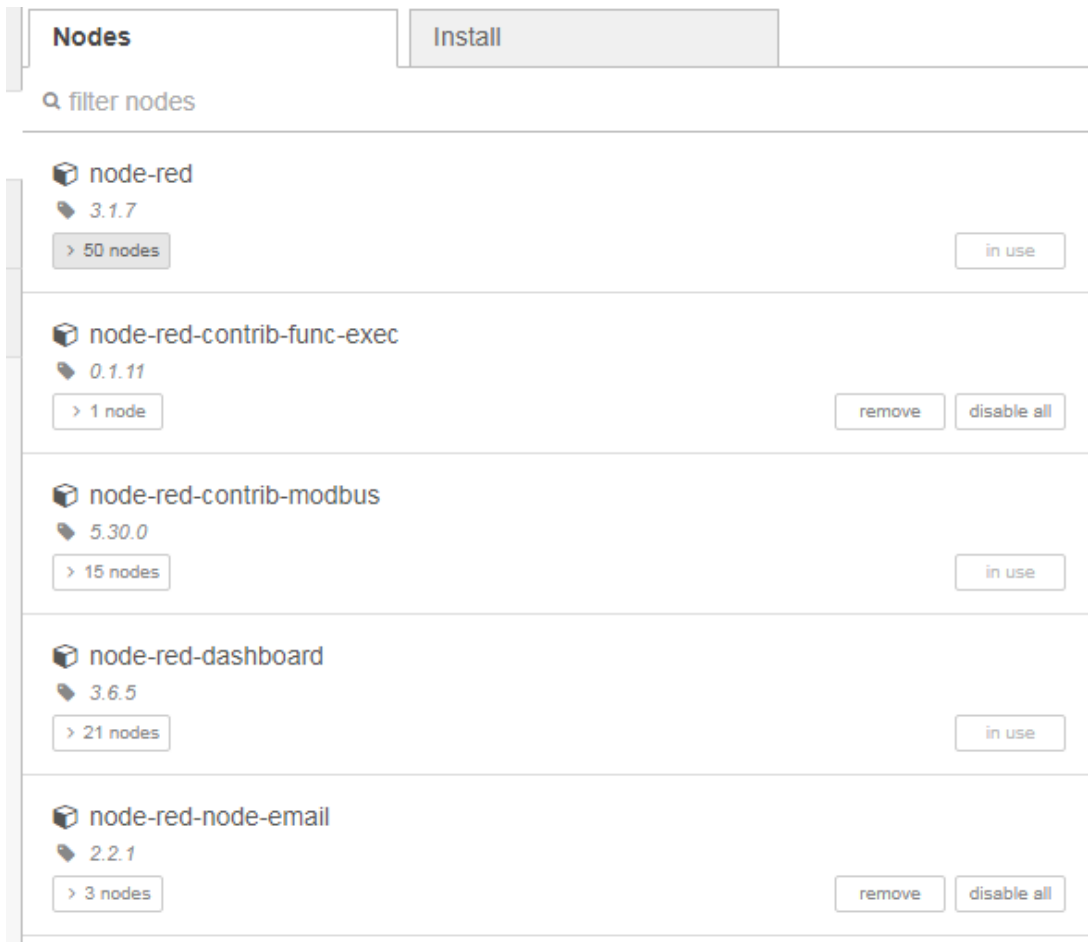
INSTALL node-red-contrib-modbus

Open menu "Manage palette" and select tab install. Enter node-red-contrib-modbus in the search field. You should see an similar screen:



Click on Install. We have installed this component already.

We have additionally installed the components node-red-dashboard for creating a simple UI



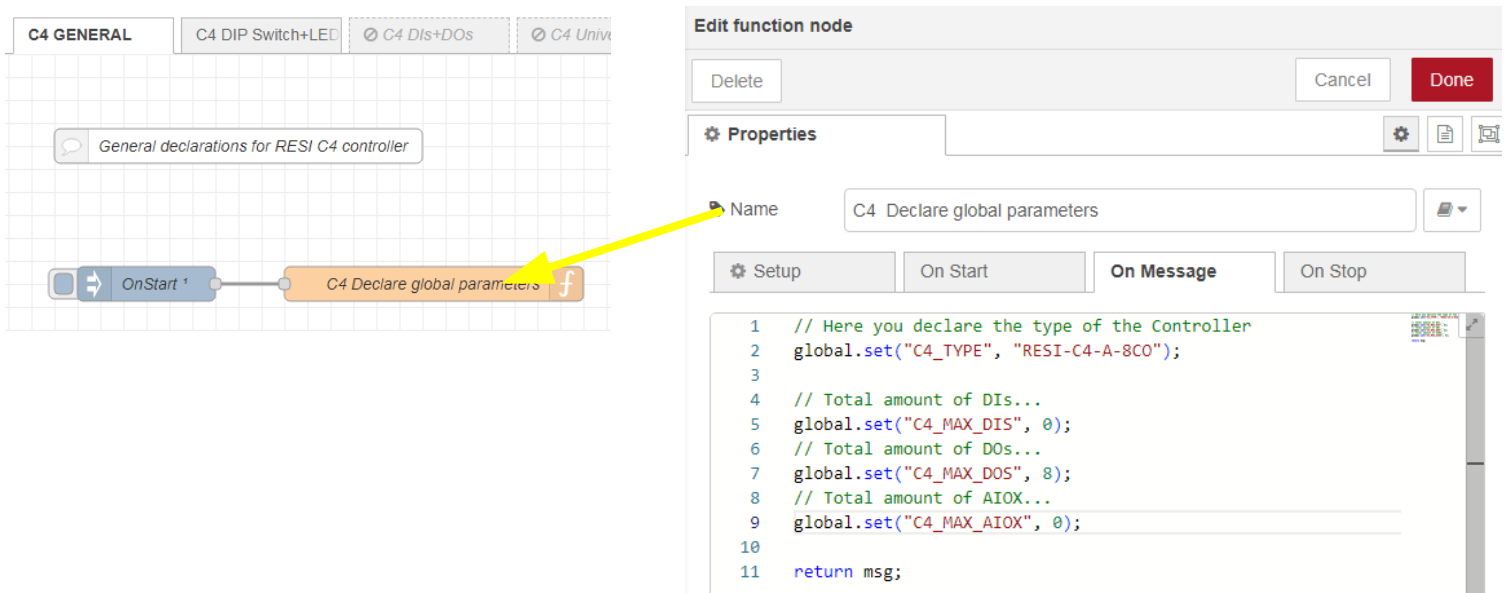
First NodeRED® flow for RESI-C4-8CO controller

First Flow

Now we create our first flow C4 GENERAL. Import this flow:

```
[{"id":"6f86f58cac7ec48d","type":"tab","label":"C4 GENERAL","disabled":false,"info":"","env":[]}, {"id":"48bcbb8a512940c3","type":"comment","z":"6f86f58cac7ec48d","name":"General declarations for RESI C4 controller","info":"","x":220,"y":80,"wires":[]}, {"id":"d6308c2bd6764109","type":"inject","z":"6f86f58cac7ec48d","name":"OnStart","props":[{"p":"payload"}], {"p":{"topic":"vt","vt":"str"},"repeat":"","crontab":"","once":true,"onceDelay":0,"topic":"","payload":"","payloadType":"date","x":140,"y":200,"wires":[["020af8fc1079e4c5"]]}, {"id":"020af8fc1079e4c5","type":"function","z":"6f86f58cac7ec48d","name":"C4 Declare global parameters","func":"// Here you declare the type of the Controller\\nglobal.set(\"C4_TYPE\", \"RESI-C4-A-8CO\");\\n\\n// Total amount of DIs...\\nglobal.set(\"C4_MAX_DIS\", 0);\\n// Total amount of DOs...\\nglobal.set(\"C4_MAX_DOS\", 8);\\n// Total amount of AIOX...\\nglobal.set(\"C4_MAX_AIOX\", 0);\\n\\nreturn msg;","outputs":0,"timeout":0,"noerr":0,"initialize":"","finalize":"","libs":[],"x":390,"y":200,"wires":[]}]
```

Your result should look like this. Double click on the function node to see the java code.



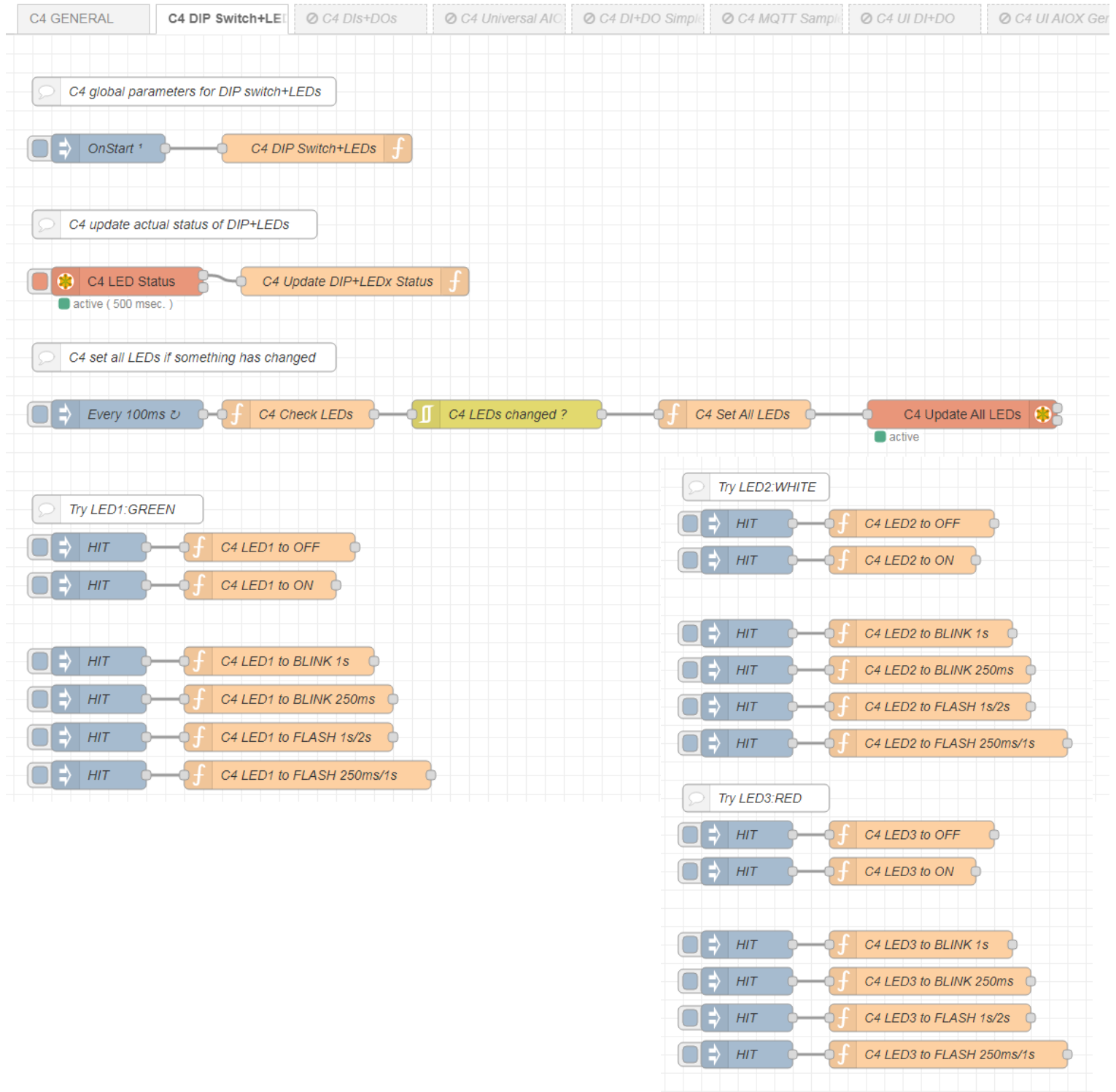
We use the `global.set(<Name>,<Value>)` function to define some global parameters for all other nodes. After you have successfully deployed the flow and you select the context menu and refresh the entries, you will see under the section global the new variables C4_TYPE, C4_MAX_DIS, C4_MAX_DOS and C4_MAX_AIOX.



NodeRED® flow to update RESI-C4-8CO DIP switch+LEDs

Read the DIP switch and set the LEDs from RESI-C4 controller

The flow should look like this:



This flow will read the DIP Switch and LED status every 500ms via [dev/ttyACM1](#) serial interface. Also it will set a new state for the LEDs, if you HIT the corresponding event triggers. Check the function nodes, what we have programmed!



NodeRED® flow to update RESI-C4-8CO DIP switch+LEDs

Read the DIP switch and set the LEDs from RESI-C4 controller

To run this sample flow, you will need to define the serial interface (in the section configuration node). Use this parameters for the **dev/ttyACM1** USB interface of our C4 controller. The settings of the baud rate are not relevant, due to the fact that the device is connected via USB to LINUX.

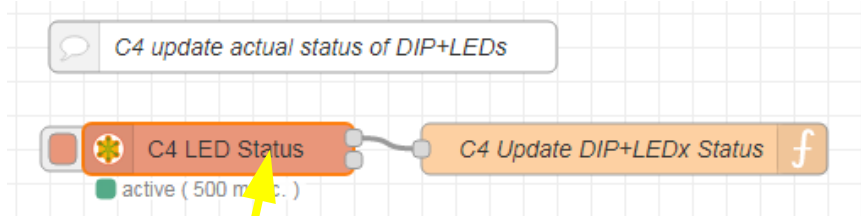
The screenshot shows the configuration for a 'modbus-client' node in NodeRED. The 'Serial port' is set to '/dev/ttyACM1'. The 'Serial type' is 'RTU-BUFFERD', 'Baud rate' is '9600', 'Data Bits' is '8', 'Stop Bits' is '1', and 'Parity' is 'None'. The 'Connection delay (ms)' is '1'. The 'Unit-Id' is '2', 'Timeout (ms)' is '1000', and 'Reconnect on timeout' is checked with a 'Reconnect timeout (ms)' of '2000'. The node is 'Enabled' and has '15' instances. The configuration is applied to 'On all flows'.



NodeRED® flow to update RESI-C4-8CO DIP switch+LEDs

Read the DIP switch and set the LEDs from RESI-C4 controller

Let's concentrate on the readout of the DIP switch+LED status first. The first node is a MODBUS/RTU read node, with a poll rate of 500ms.



Edit Modbus-Read node

Delete Cancel Done

Properties

Settings | Optionals

Name: C4 LED Status

Topic: Topic

Unit-Id: 1

FC: FC 3: Read Holding Registers

Address: 65500

Quantity: 20

Poll Rate: 500 millisecond(s)

Delay to activate input:

Server: modbus-serial@/dev/ttyACM1:9600

This node read every 500ms the MODBUS/RTU 20 holding registers starting at index 65500.

When you look into our ASCII+MODBUS register document [RESI-L-C4-A-8CO-xGB-MODBUS+ASCII-EN.pdf](#) you will find the registers starting with I:65500 containing the DIP switch, and then for every LED color some registers with the current status.

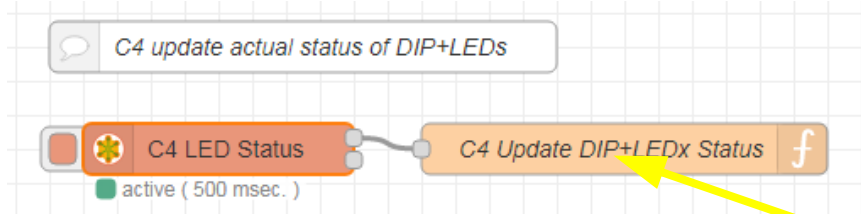
DIP SWITCH STATUS					
DIP SWITCH	3x65501 4x65501 65500	????			UINT16 R/O
Returns the actual setting of the Dip switches. Bit 0: DIP Switch 1 (=0:OFF, =1:ON) Bit 1: DIP Switch 2 (=0:OFF, =1:ON) Bit 2: DIP Switch 3 (=0:OFF, =1:ON) Bit 3: DIP Switch 4 (=0:OFF, =1:ON) Bit 4: DIP Switch 5 (=0:OFF, =1:ON) Bit 5: DIP Switch 6 (=0:OFF, =1:ON) Bit 6: DIP Switch 7 (=0:OFF, =1:ON) Bit 7: DIP Switch 8 (=0:OFF, =1:ON)					
LED1:GREEN					
LED1:GREEN STATE	3x65502 4x65502 65501	????	1:SET TO ON		UINT16 R/W NO
State of LED:????					
(C) Copyright 2024 by RESI Informatik & Automation GmbH & DI HC Sigl.MSc Page 29 of 161 13.04.2024 18:40:57					
RESI Configurator RESI-C4-8CO-xAIOX-V1007 RESI-C4-xxx MB DIP+LEDs					
Returns the actual state of the LED Writing to this register will set a new state for the LED 0: Switch LED permanent OFF 1: Switch LED permanent ON 2: Invert last state of LED 3: Start symmetrical blinking of LED with TIME1 ON and TIME1 OFF 4: Start asymmetrical flashing of LED with TIME1 ON and TIME2 OFF 5: Start one time pulse of LED with TIME1 ON and Infrate OFF					
LED1:GREEN TIME1	3x65503 4x65503 65502	????	1000		UINT16 R/W YES
Returns the actual time1 for blink, flash and pulse ON time in Milliseconds Writing to this register sets a new time in the range 20-65534ms Actual time 1 in ms:0					
LED1:GREEN TIME2	3x65504 4x65504 65503	????	2000		UINT16 R/W YES
Returns the actual time2 for blink and flash OFF time in Milliseconds Writing to this register sets a new time in the range 20-65534ms Actual time 2 in ms:0					
LED2:WHITE					
LED2:WHITE STATE	3x65505 4x65505 65504	????	1:SET TO ON		UINT16 R/W NO
State of LED:????					
Returns the actual state of the LED Writing to this register will set a new state for the LED 0: Switch LED permanent OFF 1: Switch LED permanent ON 2: Invert last state of LED 3: Start symmetrical blinking of LED with TIME1 ON and TIME1 OFF 4: Start asymmetrical flashing of LED with TIME1 ON and TIME2 OFF 5: Start one time pulse of LED with TIME1 ON and Infrate OFF					



NodeRED® flow to update RESI-C4-8CO DIP switch+LEDs

Read the DIP switch and set the LEDs from RESI-C4 controller

Let's concentrate on the readout of the DIP switch+LED status first. The first node is a MODBUS/RTU read node, with a poll rate of 500ms.



Double click on the function block to open the JavaScript editor window.

We have now the result of the MODBUS read in the message payload as an array with 20 elements starting with index 0.

Therefore `msg.payload[0]` represents the 16-bit content of the holding register I:65500 (The DIP switch). So we save this in the global variable named `C4_DIP_Actual`. You can use this value in every flow with `global.get("C4_DIP_Actual")`.

`msg.payload[1]` to `msg.payload[3]` represent the contents of the holding registers I:65501 to I:65503 representing the GREEN LED status.

`msg.payload[4]` to `msg.payload[6]` represent the contents of the holding registers I:65504 to I:65506 representing the WHITE LED status.

`msg.payload[7]` to `msg.payload[9]` represent the contents of the holding registers I:65507 to I:65509 representing the RED LED status.

`msg.payload[10]` to `msg.payload[12]` represent the contents of the holding registers I:65510 to I:65512 representing the YELLOW LED status. But this status we do not save, because we cannot control this LED. It represents the current status of the digital outputs. If one output is on, the LED is ON too!

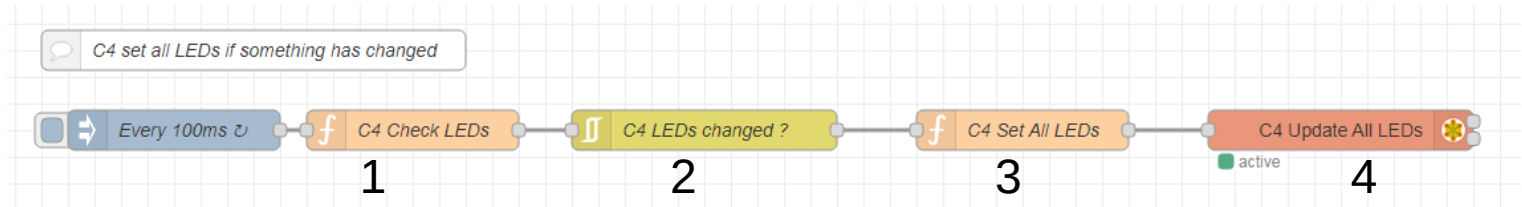
Again you can access the current LED state from every flow with the global variables `C4_LEDx_State`, `C4_LEDx_Time1`, `C4_LEDx_Time2`

```
1 // 4x65501, I:65500
2 // Actual state of DIP switch
3 global.set("C4_DIP_Actual", msg.payload[0]);
4
5 // 4x65502, I:65501
6 // LED1:GREEN: Actual state
7 global.set("C4_LED1_State", msg.payload[1]);
8 // 4x65503, I:65502
9 // LED1:GREEN: Actual Time1 in ms
10 global.set("C4_LED1_Time1", msg.payload[2]);
11 // 4x65504, I:65503
12 // LED1:GREEN: Actual Time2 in ms
13 global.set("C4_LED1_Time2", msg.payload[3]);
14
15 // 4x65505, I:65504
16 // LED2:WHITE: Actual state
17 global.set("C4_LED2_State", msg.payload[4]);
18 // 4x65506, I:65505
19 // LED2:WHITE: Actual Time1 in ms
20 global.set("C4_LED2_Time1", msg.payload[5]);
21 // 4x65507, I:65506
22 // LED2:WHITE: Actual Time2 in ms
23 global.set("C4_LED2_Time2", msg.payload[6]);
24
25 // 4x65508, I:65507
26 // LED3:RED: Actual state
27 global.set("C4_LED3_State", msg.payload[7]);
28 // 4x65509, I:65508
29 // LED3:RED: Actual Time1 in ms
30 global.set("C4_LED3_Time1", msg.payload[8]);
31 // 4x65510, I:65509
32 // LED3:RED: Actual Time2 in ms
33 global.set("C4_LED3_Time2", msg.payload[9]);
34
35 return msg;
```


NodeRED® flow to update RESI-C4-8CO DIP switch+LEDs

Read the DIP switch and set the LEDs from RESI-C4 controller

The next flow will update the LEDs via MODBUS/RTU write holding register command in the C4, if you write to the global variables.



Edit function node 1

Delete Cancel Done

Properties

Name: C4 Check LEDs

Setup On Start On Message On Stop

```
1 // generate message for all LEDs and all LED parameters...
2 var msg1={ topic: "C4_LEDs", payload:[] };
3
4 msg1.payload[0] = global.get("C4_LED1_State");
5 msg1.payload[1] = global.get("C4_LED1_Time1");
6 msg1.payload[2] = global.get("C4_LED1_Time2");
7
8 msg1.payload[3] = global.get("C4_LED2_State");
9 msg1.payload[4] = global.get("C4_LED2_Time1");
10 msg1.payload[5] = global.get("C4_LED2_Time2");
11
12 msg1.payload[6] = global.get("C4_LED3_State");
13 msg1.payload[7] = global.get("C4_LED3_Time1");
14 msg1.payload[8] = global.get("C4_LED3_Time2");
15
16 return msg1 ;
```

Every 100ms we build a new message. We add the current content of the global variables for the three LEDs into the payload. Then we return the new created message to the flow for the next node.

Edit filter node 2

Delete Cancel Done

Properties

Mode: block unless value changes

Property: msg.payload

Apply mode separately for each

msg.topic

Name: C4 LEDs changed ?

This node will block the execution in this flow as long as no value has changed in the message.



NodeRED® flow to update RESI-C4-8CO DIP switch+LEDs

```
// create message for
// MODBUS WRITE MULTIPLE HOLDING REGISTERS 3
let C4_LEDs=[ 0,0,0, 0,0,0, 0,0,0];

// 4x65502, I:65501
// LED1:GREEN: Actual state
C4_LEDs[0]=global.get("C4_LED1_State");
// 4x65503, I:65502
// LED1:GREEN: Actual Time1 in ms
C4_LEDs[1]=global.get("C4_LED1_Time1");
// 4x65504, I:65503
// LED1:GREEN: Actual Time2 in ms
C4_LEDs[2]=global.get("C4_LED1_Time2");

// 4x65505, I:65504
// LED2:WHITE: Actual state
C4_LEDs[3]=global.get("C4_LED2_State");
// 4x65506, I:65505
// LED2:WHITE: Actual Time1 in ms
C4_LEDs[4]=global.get("C4_LED2_Time1");
// 4x65507, I:65506
// LED2:WHITE: Actual Time2 in ms
C4_LEDs[5]=global.get("C4_LED2_Time2");

// 4x65508, I:65507
// LED3:RED: Actual state
C4_LEDs[6]=global.get("C4_LED3_State");
// 4x65509, I:65508
// LED3:RED: Actual Time1 in ms
C4_LEDs[7]=global.get("C4_LED3_Time1");
// 4x65510, I:65509
// LED3:RED: Actual Time2 in ms
C4_LEDs[8]=global.get("C4_LED3_Time2");

msg.payload = {
  value: [
    C4_LEDs[0], C4_LEDs[1], C4_LEDs[2],
    C4_LEDs[3], C4_LEDs[4], C4_LEDs[5],
    C4_LEDs[6], C4_LEDs[7], C4_LEDs[8]
  ],
  // WRITE MULTIPLE REGISTERS
  'fc': 16,
  // INTERNAL UnitID of C4
  'unitid': 1,
  // START ADDRESS: 4x65502,I:65501
  'address': 65501,
  // 3 LEDS with 3 values each...
  'quantity': 9
}
return msg;
```

In this function block we first build an array of 9 elements to keep the new state of the global variables. The array is called C4_LEDs.

Then we want to prepare a message for the MODBUS/RTU flex-write node. Therefore our message must have the correct format.

'fc': 16 stands for the MODBUS function code 16 which means write holding registers.

'unitid': 1 stands for the MODBUS unit ID. For our controllers always 1.

'address': 65501 stands for the first MODBUS holding register index (Starting with base=0), which we want to write new values into.

'quantity': 9 defines, that we want to write to 9 consecutive holding registers.



NodeRED® flow to update RESI-C4-8CO DIP switch+LEDs

Edit Modbus-Flex-Write node

4

Delete Cancel Done

Properties

Settings Options

Name: C4 Update All LEDs

Server: modbus-serial@/dev/ttyACM1:9600

Delay to activate input

The flex write block executes the MODBUS command defined in the incoming message. We have only to select the correct serial interface dev/ttyACM1

Edit Modbus-Flex-Write node

Delete Cancel Done

Properties

Settings Options

Empty msg on Modbus fail

Keep Msg Properties

Show Activities

Show Errors

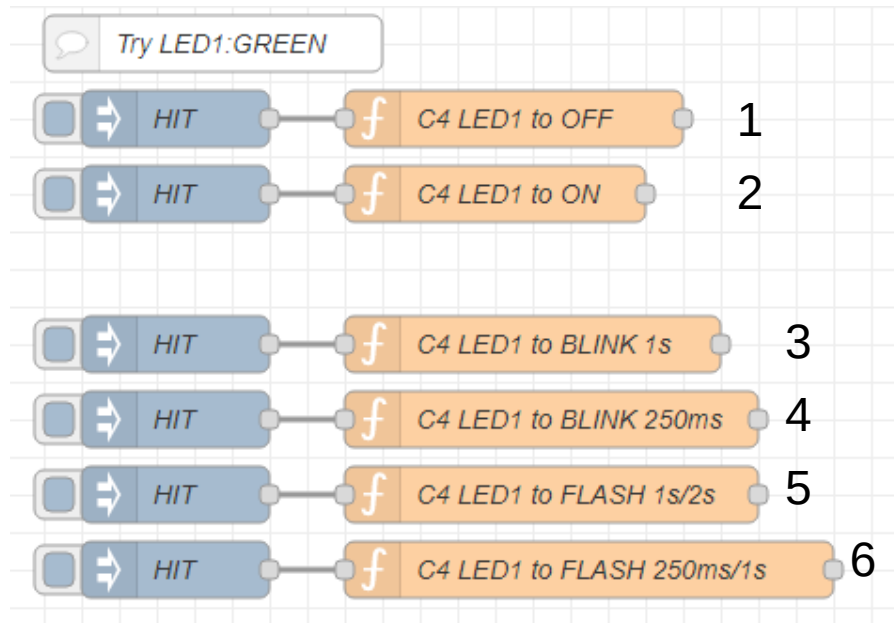
Show Warnings



NodeRED® flow to update RESI-C4-8CO DIP switch+LEDs

Control the LEDs with Node-RED

The next flow will update the status of the green LED. The white and red LEDs work similar. We use the event object titled HIT to indicate, that whenever you press this node with the mouse you will trigger the corresponding action to test easily every LED state interactively. But the deeper sense is that you learn, how writing to global variables will trigger MODBUS write actions.



Every 100ms we build a new message. We add the current content of the global variables for the three LEDs into the payload. Then we return the new created message to the flow for the next node.

Edit function node

Delete 1 Cancel Done

Properties

Name: C4 LED1 to OFF

Setup On Start **On Message** On Stop

```
1 global.set("C4_LED1_State",0);
2
3 return msg;
```

Setting the global variable C4_LED1_State to 0 will update the LED via MODBUS write and switch the LED OFF!

Name: C4 LED1 to ON 2

Setup On Start **On Message** On Stop

```
1 global.set("C4_LED1_State",1);
2
3 return msg;
```

Setting the global variable C4_LED1_State to 1 will update the LED via MODBUS write and switch the LED ON!



NodeRED® flow to update RESI-C4-8CO DIP switch+LEDs

Name: C4 LED1 to BLINK 1s **3**

Setup On Start On Message On Stop

```
1 global.set("C4_LED1_State",3);
2 global.set("C4_LED1_Time1",1000);
3
4 return msg;
```

Setting the global variable C4_LED1_State to 3 will update the LED via MODBUS write. The LED will flash cyclically with 1000ms ON and 1000ms OFF interval!

Name: C4 LED1 to BLINK 250ms **4**

Setup On Start On Message On Stop

```
1 global.set("C4_LED1_State",3);
2 global.set("C4_LED1_Time1",250);
3
4 return msg;
```

Setting the global variable C4_LED1_State to 3 will update the LED via MODBUS write. The LED will flash cyclically with 250ms ON and 250ms OFF interval!

Edit function node **5**

Delete Cancel Done

Properties

Name: C4 LED1 to FLASH 1s/2s

Setup On Start On Message On Stop

```
1 global.set("C4_LED1_State",4);
2 global.set("C4_LED1_Time1",1000);
3 global.set("C4_LED1_Time2",2000);
4
5 return msg;
```

Setting the global variable C4_LED1_State to 4 will update the LED via MODBUS write. The LED will flash cyclically with 1000ms ON and 2000ms OFF interval!

Edit function node **6**

Delete Cancel Done

Properties

Name: C4 LED1 to FLASH 250ms/1s

Setup On Start On Message On Stop

```
1 global.set("C4_LED1_State",4);
2 global.set("C4_LED1_Time1",250);
3 global.set("C4_LED1_Time2",1000);
4
5 return msg;
```

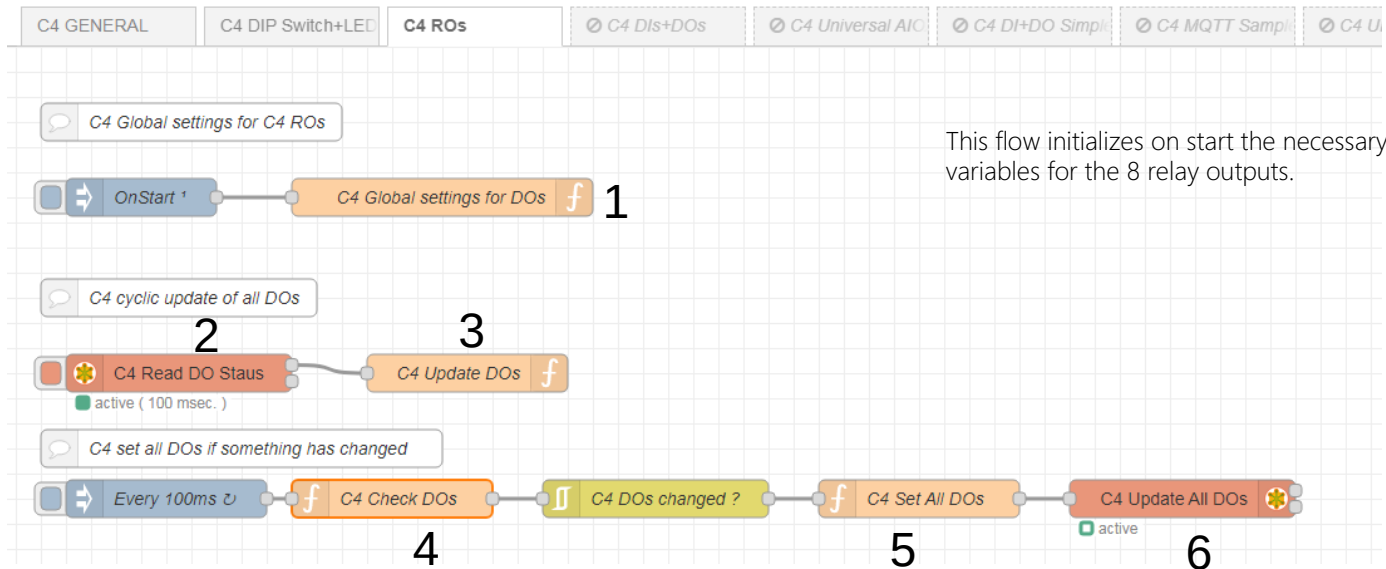
Setting the global variable C4_LED1_State to 4 will update the LED via MODBUS write. The LED will flash cyclically with 250ms ON and 1000ms OFF interval!



NodeRED® flow to update RESI-C4-8CO relay outputs

Set C4 relay outputs with Node-RED

The next flow will cyclically update the relay outputs to the new status. Therefore we explain every node.



This flow initializes on start the necessary global variables for the 8 relay outputs.

```

1  var i;
2
3  // n ROs ....
4  for (i=0;i<global.get("C4_MAX_DOS");i++)
5  {
6    global.set("C4_DO"+String(i+1).padStart(2, '0'), 0);
7    global.set("C4_DO"+String(i+1).padStart(2, '0')+"_Actual", 0);
8  }
9
10 return msg;

```

Create the global variables C4_DO01 to C4_DO08 representing the new status for the digital outputs. And also creating C4_DO01_Actual to C4_DO08_Actual representing the current state of the digital outputs

This flow will read every 100ms the actual status of the 8 digital outputs and store this status in the global variables C4_DOxx_Actual. The result of the MODBUS read of one holding register is a message with one 16-bit value as payload. We then extract every bit out of this word to save the current status for the 8 relay outputs.

```

1  var i;
2  let Word=0,Bit=0,V=0;
3
4  // Register list for C4-A-8CO
5  // 4x10001,I:10000: DIGITAL OUTPUTS D01-D08
6
7  // n ROs
8  for (i = 0; i < global.get("C4_MAX_DOS"); i++) {
9    Word=~(i/16);
10   Bit=~(i%16);
11   V=(msg.payload[Word]&(1<<Bit)) ? 1:0;
12   global.set("C4_DO"+String(i+1).padStart(2, '0')+"_Actual",V);
13 }
14
15 return msg ;

```



NodeRED® flow to update RESI-C4-8CO relay outputs

```
1 // create message with all DOs
2 var msg1={ topic: "C4_DOs", payload:[] };
3 var i
4
5 // n ROs
6 for (i = 0; i < global.get("C4_MAX_DOs"); i++) {
7   msg1.payload[i] = global.get("C4_DO" + String(i+1).padStart(2, '0'));
8 }
9
10 return msg1 ;
```

4

Again we prepare an new message containing the current status of all 8 digital outputs from the global variables C4_DOxx as payload. The next node C4 DOs changed? will wait as long as there is no change in the digital outputs. As soon as we change one of the global variables this node will trigger the MODBUS write to the holding register.

```
var i;

// for C4-A-8CO controller ...
if (global.get("C4_TYPE")=="RESI-C4-A-8CO")
{
  // Our controller offers 8 Relay outputs
  // therefore we need 1 16 Bit registers
  // 4x10001,I:10000: OUTPUTS DO1-DO8
  let C4_DOS=[ 0 ];
  let Word=0,Bit=0;

  // 8RO controller
  C4_DOS[0]=0;

  // Check for new value of C4_DOxx
  for (i=0;i<global.get("C4_MAX_DOs");i++)
  {
    Word=~(i/16);
    Bit=~(i%16);
    if (global.get("C4_DO" +String(i+1).padStart(2, '0'))==1)
    {
      C4_DOS[Word]=C4_DOS[Word]|(1<<Bit);
    }
  }

  msg.payload = {
    'value': [ C4_DOS[0] ],
    // WRITE MULTIPLE HOLDING REGISTERS
    'fc': 16,
    // Our C4 MODBUS UnitID
    'unitid': 1,
    // 4x10001,I:10000: OUTPUTS DO1-DO8
    'address': 10000 ,
    // for the 8 DOs or 1 registers...
    'quantity': 1
  }
}
else
{
  msg.payload=null;
}
return msg;
```

5

In the function node C4 Set all DOs we prepare the message for a MODBUS flex write to a holding register.

First we check, if the correct controller is defined. Then we build out of the 8 global variables C4_DOxx a variable, where every bit stands for one digital output.

Then we want to prepare a message for the MODBUS/RTU flex-write node. Therefore our message must have the correct format.

'fc': 16 stands for the MODBUS function code 16 which means write holding registers.

'unitid': 1 stands for the MODBUS unit ID. For our controllers always 1.

'address': 10000 stands for the first MODBUS holding register index (Starting with base=0), where we want to write the new output state.

'quantity': 1 defines, that we want to write only one holding register.

Edit Modbus-Flex-Write node

Delete 6 Cancel Done

⚙️ 📄 🖨️

Settings Options

Name C4 Update All DOs

Server modbus-serial@/dev/ttyACM1:9600

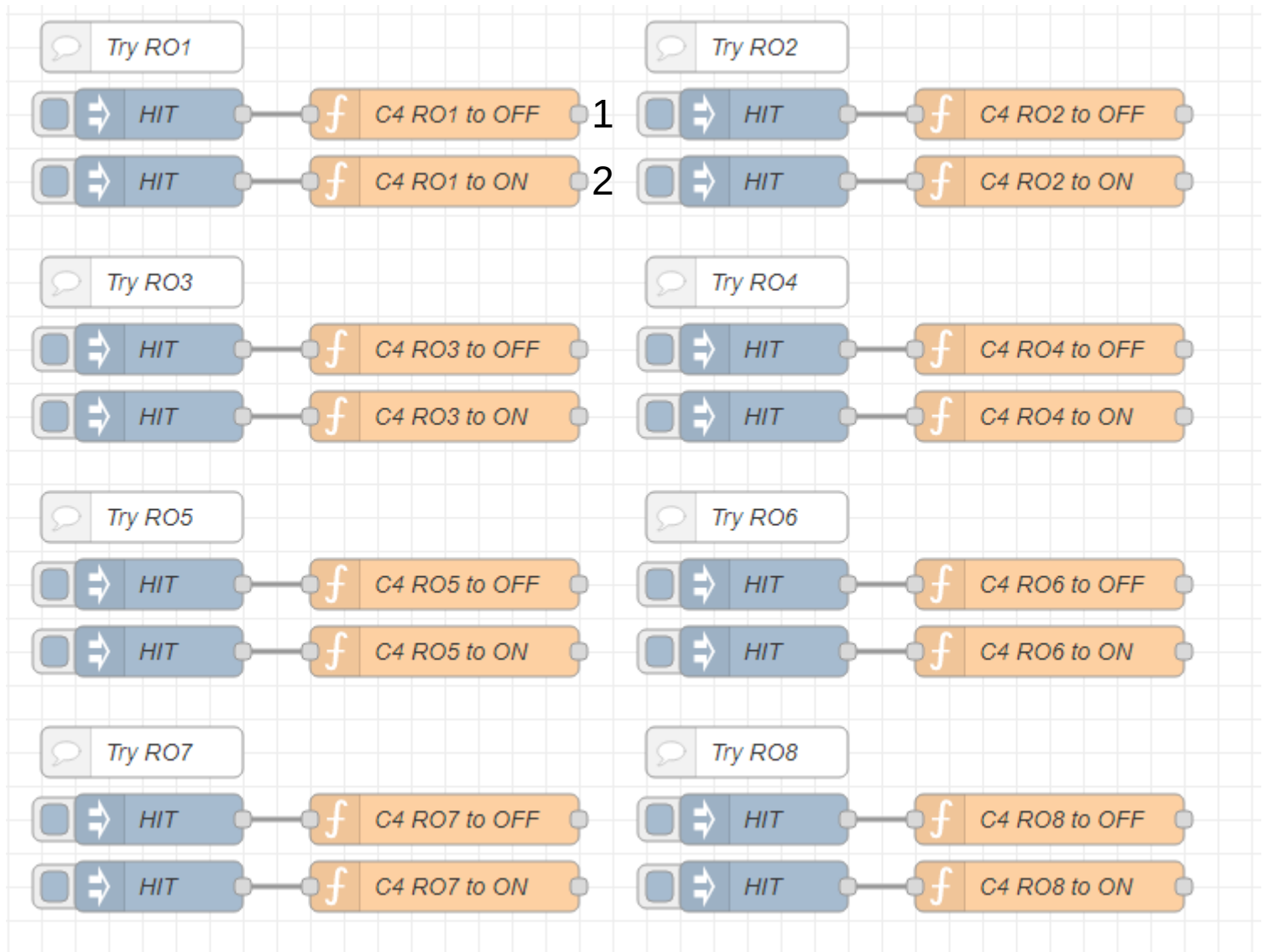
⏸ Delay to activate input



NodeRED® flow to test RESI-C4-8CO relay outputs

Testing the relay outputs with Node-RED

The next flows will switch a certain relay output to ON or OFF respectively. We use the event object titled HIT to indicate, that whenever you press this node with the mouse you will trigger the corresponding action to test easily every relay output interactively. But the deeper sense is that you learn, how writing to global variables will trigger MODBUS write actions in other flows.



```
1 global.set("C4_DO01",0);  
2  
3 return msg; 1
```

```
1 global.set("C4_DO01",1);  
2  
3 return msg; 2
```

Setting the global variable C4_DOxx to 0 will switch the relay output to OFF, writing 1 to this variable will set the relay output to ON!



NodeRED® flow to read/write relay outputs to MQTT

Read/Write the relay outputs from RESI-C4 controller to MQTT

Now we create a new flow C4 MQTT. Import this flow:

```
[{"id":"b445cbb5f08a2222","type":"tab","label":"C4 MQTT","disabled":false,"info":"","env":{}}, {"id":"a580513f97b702c6","type":"change","z":"b445cbb5f08a2222","name":"C4 Get D001","rules":[{"t":"set","p":"payload","pt":"msg","to":"C4_D001","tot":"global"}],"action":"","property":"","from":"","to":"","reg":false,"x":320,"y":80,"wires":[["37852fe763aeb3eb"]]}, {"id":"829cc13c9b3f65c0","type":"inject","z":"b445cbb5f08a2222","name":"Every 10ms","props":{"p":"payload"}, {"p":"topic","vt":"str"},"repeat":"0.01","crontab":"","once":false,"onceDelay":0.1,"topic":"","payload":"","payloadType":"date","x":130,"y":80,"wires":[["a580513f97b702c6"]]}, {"id":"8e48b7566a7ae908","type":"function","z":"b445cbb5f08a2222","name":"C4 Create MQTT D001 data","func":"var msg1={ \n topic:\\RESI_C4/DigitalOutputs/C4_D001\\, \n payload: msg.payload \n}\\n\\nreturn msg1;","outputs":1,"timeout":0,"noerr":0,"initialize":"","finalize":"","libs":[],"x":740,"y":80,"wires":[["26286143d8f50be9"]]}, {"id":"37852fe763aeb3eb","type":"rbe","z":"b445cbb5f08a2222","name":"Wait for Change","func":"rbe","gap":"","start":"","inout":"out","septopics":true,"property":"payload","topi":"topic","x":500,"y":80,"wires":[["8e48b7566a7ae908"]]}, {"id":"9b938ca18f7a3c07","type":"comment","z":"b445cbb5f08a2222","name":"Send D01 on change to MQTT","info":"","x":170,"y":40,"wires":[]}, {"id":"26286143d8f50be9","type":"mqtt out","z":"b445cbb5f08a2222","name":"RESI MQTT","topic":"","qos":"0","retain":"true","respTopic":"","contentType":"","userProps":"","correl":"","expiry":"","broker":"","url":"mqtt://192.168.1.100","x":950,"y":80,"wires":[]}, {"id":"dd7df4960ebd3b9","type":"mqtt in","z":"b445cbb5f08a2222","name":"C4 Wait for MQTT D002","topic":"RESI_C4/DigitalOutputs/C4_D002","qos":"2","datatype":"auto-detect","broker":"","url":"mqtt://192.168.1.100","x":150,"y":240,"wires":[["0a47ea6ef2f263a3"]]}, {"id":"0a47ea6ef2f263a3","type":"function","z":"b445cbb5f08a2222","name":"C4 Set D002","func":"if (msg.payload == 0 || msg.payload == 1) {\n global.set(\\C4_D002\\, msg.payload);\\n\\nreturn msg;","outputs":1,"timeout":0,"noerr":0,"initialize":"","finalize":"","libs":[],"x":350,"y":240,"wires":[]}, {"id":"8da497303a63d932","type":"mqtt in","z":"b445cbb5f08a2222","name":"C4 Wait for MQTT D0x","topic":"RESI_C4/DigitalOutputs/#","qos":"2","datatype":"auto-detect","broker":"","url":"mqtt://192.168.1.100","x":150,"y":380,"wires":[["e38c5fafd08b05bf"]]}, {"id":"e38c5fafd08b05bf","type":"function","z":"b445cbb5f08a2222","name":"C4 get MQTT D0x","func":"if (msg.topic.startsWith(\\RESI_C4/DigitalOutputs/\\)) \\n \\n // retrieve the last word in the topic...\\n let DOx=msg.topic.split(\\//).pop();\\n msg.topic=DOx;\\n\\nreturn msg;","outputs":1,"timeout":0,"noerr":0,"initialize":"","finalize":"","libs":[],"x":410,"y":380,"wires":[]}, {"id":"19f5f6c9390cc2f4","type":"function","z":"b445cbb5f08a2222","name":"C4 Set D0x","func":"let ok=false;\\n\\nif (msg.topic.startsWith(\\C4_DO\\)) \\n\\n let DOx=msg.topic.substring(5,10);\\n if (Number.isInteger(parseInt(DOx))) \\n let DOx=parseInt(DOx);\\n if (DOx>=1 && DOx<global.get(\\C4_MAX_DOS\\)) \\n (\\n ok=true;\\n \\n if ((msg.payload == 0 || msg.payload == 1) && ok) \\n (\\n var DOXNAME=\\C4_DO\\+String(DOx).padStart(2, '0');\\n global.set(DOXNAME, msg.payload);\\n \\n \\n)\\n\\nreturn msg;","outputs":1,"timeout":0,"noerr":0,"initialize":"","finalize":"","libs":[],"x":610,"y":380,"wires":[]}, {"id":"6d315be4070494d3","type":"inject","z":"b445cbb5f08a2222","name":"Every 10ms","props":{"p":"payload"}, {"p":"topic","vt":"str"},"repeat":"0.01","crontab":"","once":false,"onceDelay":0.1,"topic":"","payload":"","payloadType":"date","x":130,"y":500,"wires":[["781b7bbc8e88990a"]]}, {"id":"5516f4ecaba82a26","type":"function","z":"b445cbb5f08a2222","name":"C4 Create MQTT D0x data","func":"var msg1={ \n topic:\\RESI_C4/DigitalOutputs/C4_D001\\, \n payload: msg.payload[0] \n}\\n\\nvar msg2={ \n topic:\\RESI_C4/DigitalOutputs/C4_D002\\, \n payload: msg.payload[1] \n}\\n\\nvar msg3={ \n topic:\\RESI_C4/DigitalOutputs/C4_D003\\, \n payload: msg.payload[2] \n}\\n\\nvar msg4={ \n topic:\\RESI_C4/DigitalOutputs/C4_D004\\, \n payload: msg.payload[3] \n}\\n\\nvar msg5={ \n topic:\\RESI_C4/DigitalOutputs/C4_D005\\, \n payload: msg.payload[4] \n}\\n\\nvar msg6={ \n topic:\\RESI_C4/DigitalOutputs/C4_D006\\, \n payload: msg.payload[5] \n}\\n\\nvar msg7={ \n topic:\\RESI_C4/DigitalOutputs/C4_D007\\, \n payload: msg.payload[6] \n}\\n\\nvar msg8={ \n topic:\\RESI_C4/DigitalOutputs/C4_D008\\, \n payload: msg.payload[7] \n}\\n\\nreturn [ msg1, msg2, msg3, msg4, msg5, msg6, msg7, msg8 ];","outputs":8,"timeout":0,"noerr":0,"initialize":"","finalize":"","libs":[]}, {"id":"b07102cf31d3807f","type":"comment","z":"b445cbb5f08a2222","name":"C4 Dos changed ?","func":"rbe","gap":"","start":"","inout":"out","septopics":true,"property":"payload","topi":"topic","x":510,"y":500,"wires":[]}, {"id":"5516f4ecaba82a26","type":"comment","z":"b445cbb5f08a2222","name":"Send all DOx on change to MQTT","info":"","x":180,"y":460,"wires":[]}, {"id":"b07102cf31d3807f","type":"mqtt out","z":"b445cbb5f08a2222","name":"RESI MQTT","topic":"","qos":"0","retain":"true","respTopic":"","contentType":"","userProps":"","correl":"","expiry":"","broker":"","url":"mqtt://192.168.1.100","x":1130,"y":540,"wires":[]}, {"id":"781b7bbc8e88990a","type":"function","z":"b445cbb5f08a2222","name":"C4 Check DOs","func":"// create message with all DOs\\nvar msg1={ topic: \\C4_DOS\\, payload: [ ] };\\nvar i\\n\\nfor (i = 0; i < global.get(\\C4_MAX_DOS\\); i++) {\\n msg1.payload[i] = global.get(\\C4_DO\\ + String(i+1)).padStart(2, '0');\\n}\\n\\nreturn msg1;","outputs":1,"timeout":0,"noerr":0,"initialize":"","finalize":"","libs":[],"x":300,"y":500,"wires":[]}, {"id":"7d077fd70eadd07","type":"comment","z":"b445cbb5f08a2222","name":"Receive from MQTT new status for DO2","info":"","x":190,"y":180,"wires":[]}, {"id":"449b5cb5c7728702b","type":"comment","z":"b445cbb5f08a2222","name":"Receive from MQTT new status for all DOx","info":"","x":200,"y":320,"wires":[]}, {"id":"9d7784867c5e3d27","type":"mqtt broker","name":"","broker":"127.0.0.1","port":"1883","clientId":"","autoConnect":true,"usetls":false,"protocolVersion":"4","keepalive":"60","cleansession":true,"autoUnsubscribe":true,"birthTopic":"","birthQos":"0","birthRetain":"false","birthPayload":"","birthMsg":{},"closeTopic":"","closeQos":"0","closeRetain":"false","closePayload":"","closeMsg":{},"willTopic":"","willQos":"","willRetain":"false","willPayload":"","willMsg":{},"userProps":"","sessionExpiry":""}]
```


NodeRED® flow to read/write relay outputs to MQTT

HOWTO send actual status to MQTT broker

First of all we setup a MQTT server. We use MOSQUITTO. There are many installation guides, howto setup mosquitto on LINUX. After successful installation you can check the mosquitto server with. We use resimqtt as user and r4MQTT as password. Please use better and safer credentials in your installation!

```
mosquitto_sub -t RESI_C4/# -d -u resimqtt -P r4MQTT
```

This command will show all entries in the MQTT server starting with RESI_C4, which is our root.

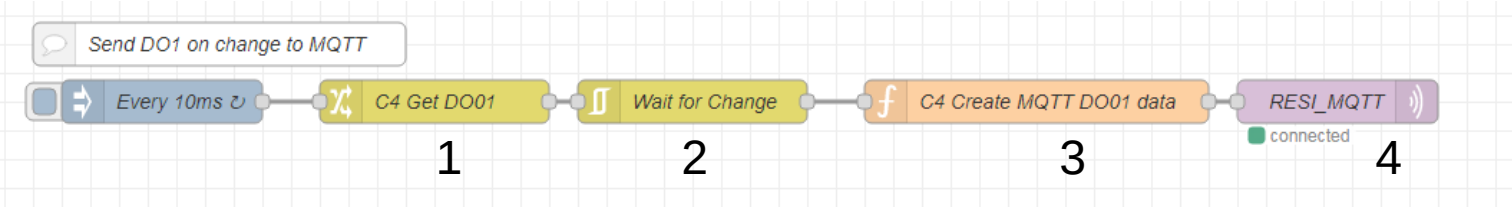
```
resi@RESI-C4:~$ mosquitto_sub -t RESI_C4/# -d -u resimqtt -P r4MQTT
Client (null) sending CONNECT
Client (null) received CONNACK (0)
Client (null) sending SUBSCRIBE (Mid: 1, Topic: RESI_C4/#, QoS: 0, Options: 0x00)
Client (null) received SUBACK
Subscribed (mid: 1): 0
Client (null) received PUBLISH (d0, q0, r1, m0, 'RESI_C4/DigitalInputs/Cx_DI1', ... (1 bytes))
1
Client (null) received PUBLISH (d0, q0, r1, m0, 'RESI_C4/DigitalInputs/C4_DI1', ... (1 bytes))
0
Client (null) received PUBLISH (d0, q0, r1, m0, 'RESI_C4/WL01/LEVEL', ... (4 bytes))
1234
Client (null) received PUBLISH (d0, q0, r1, m0, 'RESI_C4/WL01/TEMPERATURE', ... (5 bytes))
22.56
Client (null) received PUBLISH (d0, q0, r1, m0, 'RESI_C4', ... (1 bytes))
1
Client (null) sending PINGREQ
Client (null) received PINGRESP
```



NodeRED® flow to read/write relay outputs to MQTT

Set C4 relay outputs with Node-RED

The next flow will cyclically check the status of relay output DO1. If there is a change, a message is sent to MQTT server.



Edit change node 1

Delete Cancel Done

Properties

Name: C4 Get DO01

Rules

Set msg. payload to the value global. C4_DO01

Deep copy value

This node generates a message with the global variable C4_DO01.

Edit filter node 2

Delete Cancel Done

Properties

Mode: block unless value changes

Property: msg. payload

Apply mode separately for each

msg. topic

Name: Wait for Change

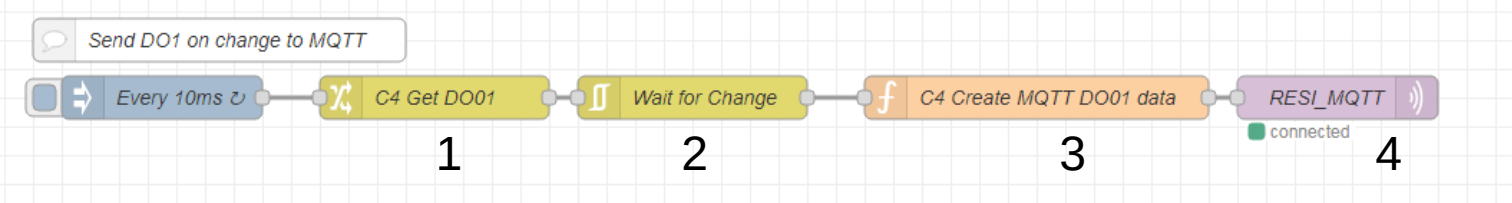
This node blocks the event to the next node as long as the content of the message has not changed.



NodeRED® flow to read/write relay outputs to MQTT

Set C4 relay outputs with Node-RED

The next flow will cyclically check the status of relay output DO1. If there is a change, a message is sent to MQTT server.



Edit function node 3

Delete Cancel Done

Properties

Name: C4 Create MQTT DO01 data

Setup On Start On Message On Stop

```
1 var msg1={
2   topic:"RESI_C4/DigitalOutputs/C4_DO01",
3   payload: msg.payload
4 }
5
6 return msg1;
```

This node prepares the message to send to the MQTT server.

Edit mqtt out node 4

Delete Cancel Done

Properties

Server: 127.0.0.1:1883

Topic: Topic

QoS: 0 Retain: true

Name: RESI_MQTT

Tip: Leave topic, qos or retain blank if you want to set them via msg properties.

This node will send the received message with the topic to the MQTT Server.



NodeRED® flow to read/write relay outputs to MQTT

Set C4 relay outputs with Node-RED

In the section of the configuration nodes you will find the MQTT broker node, which establishes the connection to your MQTT server and where you have to define your credentials. Also you can define messages in case of connection/disconnection.

The image displays three screenshots of the Node-RED MQTT broker node configuration interface, showing different tabs and settings.

Top Left Screenshot (Connection Tab): Shows the 'Edit mqtt-broker node' configuration. The 'Connection' tab is active. Fields include: Name (Name), Server (127.0.0.1), Port (1883), Protocol (MQTT V3.1.1), Client ID (Leave blank for auto generated), Keep Alive (60), and Session (Use clean session checked).

Top Right Screenshot (Security Tab): Shows the 'Edit mqtt-broker node' configuration. The 'Security' tab is active. Fields include: Username (resimqtt) and Password (masked with dots).

Bottom Screenshot (Messages Tab): Shows the 'Edit mqtt-broker node' configuration. The 'Messages' tab is active. It displays three sections for defining messages:

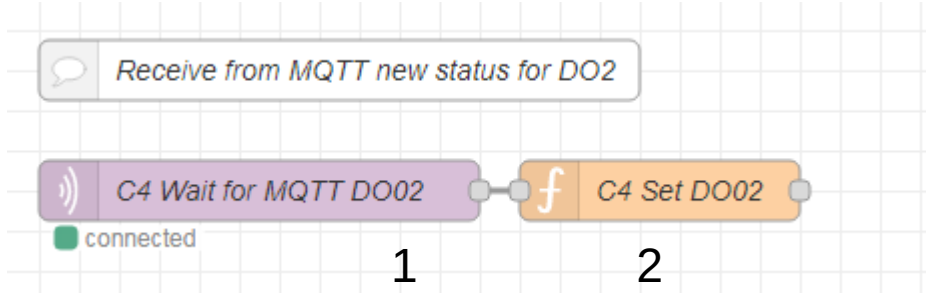
- Message sent on connection (birth message):** Topic (Leave blank to disable birth message), Retain (false), Payload (Payload), QoS (0).
- Message sent before disconnecting (close message):** Topic (Leave blank to disable close message), Retain (false), Payload (Payload), QoS (0).
- Message sent on an unexpected disconnection (will message):** Topic (Leave blank to disable will message), Retain (false), Payload (Payload), QoS (0).



NodeRED® flow to read/write relay outputs to MQTT

Receive one specific message from MQTT to switch DO2

The next flow will wait for a specific message from MQTT broker to switch relay output DO2



This screenshot shows the configuration for the MQTT node. The "Name" field is "C4 Wait for MQTT DO02". The "Server" is "127.0.0.1:1883". The "Action" is "Subscribe to single topic". The "Topic" is "RESI_C4/DigitalOutputs/C4_DO02". The "QoS" is "2". The "Output" is "auto-detect (parsed JSON object, string or buf)". There are "Delete", "Cancel", and "Done" buttons at the top. A "1" is written next to the "Cancel" button.

This node triggers an event, whenever the MQTT broker will send the message RESI_C4/DigitalOutputs/C4_DO02

The contents of the message is 0 for OFF and 1 for ON.

This screenshot shows the configuration for the function node. The "Name" field is "C4 Set DO02". There are "Delete", "Cancel", and "Done" buttons at the top. A "2" is written next to the "Cancel" button. Below the configuration, there are tabs for "Setup", "On Start", "On Message", and "On Stop". The "On Message" tab is selected, showing the following JavaScript code:

```
1 if (msg.payload == 0 || msg.payload == 1) {
2   global.set("C4_DO02", msg.payload);
3 }
4
5 return msg;
```

This node receives the message and checks if the payload is 0 or 1.

Then we set the global variable C4_DO02 to the new value.

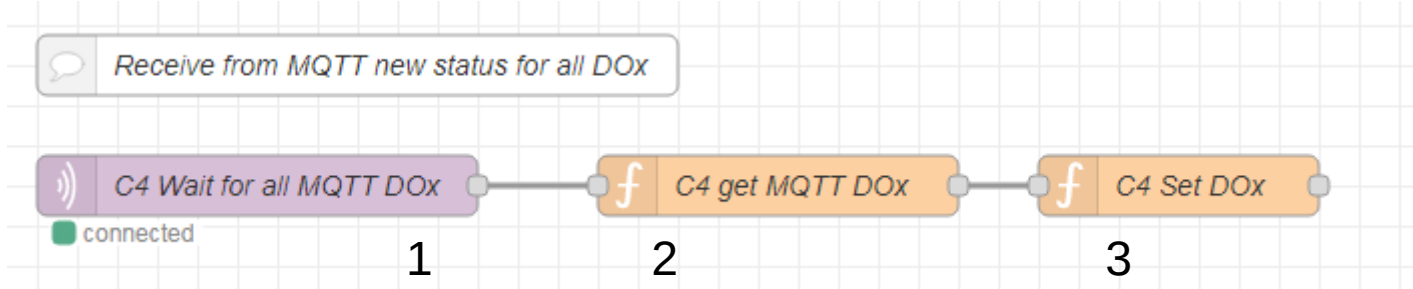
The flow C4 ROs will check the change in the set point and will write via MODBUS to the correct register to switch the digital output according to the new state.



NodeRED® flow to read/write relay outputs to MQTT

Receive more than one message from MQTT to switch all DOx

The next flow will wait for a specific message from MQTT broker to switch relay output DO2



Edit mqtt in node 1

Delete Cancel Done

Properties

Server: 127.0.0.1:1883

Action: Subscribe to single topic

Topic: RESI_C4/DigitalOutputs/#

QoS: 2

Output: auto-detect (parsed JSON object, string or buf)

Name: C4 Wait for all MQTT DOx

This node triggers an event, whenever the MQTT broker will send a message starting with RESI_C4/DigitalOutputs/

The # stands for everything

Edit function node 2

Delete Cancel Done

Properties

Name: C4 get MQTT DOx

Setup On Start On Message On Stop

```
1 if (msg.topic.startsWith("RESI_C4/DigitalOutputs/"))
2
3 // retrieve the last word in the topic...
4 let DOx=msg.topic.split("/").pop();
5 msg.topic=DOx;
6
7 return msg;
```

In this function node we check if the topic really starts with RESI_C4/DigitalOutputs/.

Only in this case we change the topic of the message to the digital output C4_DOxx



NodeRED® flow to read/write relay outputs to MQTT

Edit function node

3

Delete Cancel Done

Properties

Name C4 Set DOx

Setup On Start On Message On Stop

```
1 let ok=false;
2
3 if (msg.topic.startsWith("C4_DO"))
4 {
5   let DOnr=msg.topic.substring(5,100);
6   if (Number.isInteger(parseInt(DOnr)))
7   {
8     let DOx=parseInt(DOnr,10);
9     if (DOx>=1 && DOx<global.get("C4_MAX_DOS"))
10    {
11      | ok=true;
12    }
13    if ((msg.payload == 0 || msg.payload == 1) && ok)
14    {
15      var DOxNAME="C4_DO"+String(DOx).padStart(2, '0');
16      global.set(DOxNAME, msg.payload);
17    }
18  }
19 }
20 return msg;
```

This node checks if the topic has the name C4_DO01 to C4_DO08.

Only in this case the global variable C4_DOxx is updated with the new output state.

Again the flow on C4 ROs will then check the change and update the digital outputs via MODBUS.



NodeRED® flow to read/write relay outputs to MQTT

To test the MQTT broker use the command to view all incoming messages of your MQTT server:

```
mosquitto_sub -t RESI_C4/# -d -u resimqtt -P r4MQTT
```

To switch the digital output DO2 ON, use this command

```
mosquitto_pub -t RESI_C4/DigitalOutputs/Cx_DO2 -u resimqtt -P r4MQTT -m 1
```

To switch the digital output DO2 OFF, use this command

```
mosquitto_pub -t RESI_C4/DigitalOutputs/Cx_DO2 -u resimqtt -P r4MQTT -m 0
```

To switch the digital outputs DO1 to DO8 ON or OFF, use this command

```
mosquitto_pub -t RESI_C4/DigitalOutputs/Cx_DO1 -u resimqtt -P r4MQTT -m 1
mosquitto_pub -t RESI_C4/DigitalOutputs/Cx_DO1 -u resimqtt -P r4MQTT -m 0
...
mosquitto_pub -t RESI_C4/DigitalOutputs/Cx_DO8 -u resimqtt -P r4MQTT -m 1
mosquitto_pub -t RESI_C4/DigitalOutputs/Cx_DO8 -u resimqtt -P r4MQTT -m 0
```

As soon as NodeRED sends a new MQTT status you will receive similar logging output:

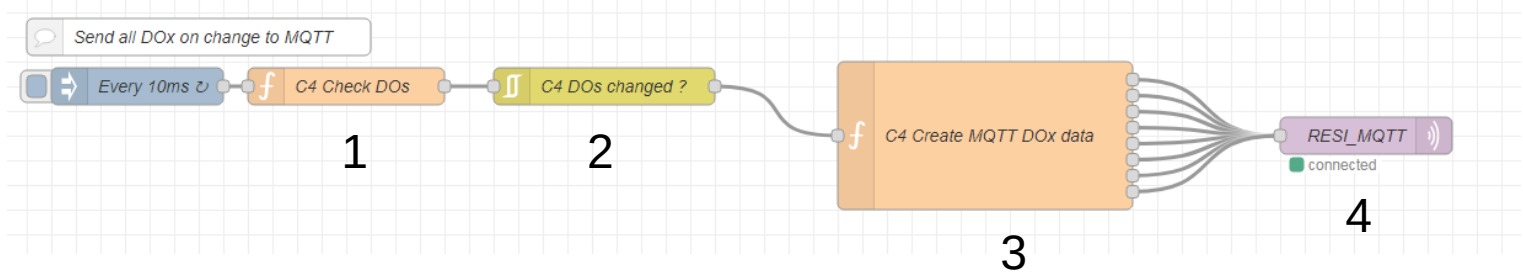
```
0
Client (null) received PUBLISH (d0, q0, r0, m0, 'RESI_C4/DigitalOutputs/C4_D004', ... (1 bytes))
0
Client (null) received PUBLISH (d0, q0, r0, m0, 'RESI_C4/DigitalOutputs/C4_D005', ... (1 bytes))
1
Client (null) received PUBLISH (d0, q0, r0, m0, 'RESI_C4/DigitalOutputs/C4_D006', ... (1 bytes))
0
Client (null) received PUBLISH (d0, q0, r0, m0, 'RESI_C4/DigitalOutputs/C4_D007', ... (1 bytes))
0
Client (null) received PUBLISH (d0, q0, r0, m0, 'RESI_C4/DigitalOutputs/C4_D008', ... (1 bytes))
0
Client (null) received PUBLISH (d0, q0, r0, m0, 'RESI_C4/DigitalOutputs/C4_D001', ... (1 bytes))
1
Client (null) received PUBLISH (d0, q0, r0, m0, 'RESI_C4/DigitalOutputs/C4_D002', ... (1 bytes))
0
Client (null) received PUBLISH (d0, q0, r0, m0, 'RESI_C4/DigitalOutputs/C4_D003', ... (1 bytes))
0
Client (null) received PUBLISH (d0, q0, r0, m0, 'RESI_C4/DigitalOutputs/C4_D004', ... (1 bytes))
0
Client (null) received PUBLISH (d0, q0, r0, m0, 'RESI_C4/DigitalOutputs/C4_D005', ... (1 bytes))
1
Client (null) received PUBLISH (d0, q0, r0, m0, 'RESI_C4/DigitalOutputs/C4_D006', ... (1 bytes))
0
Client (null) received PUBLISH (d0, q0, r0, m0, 'RESI_C4/DigitalOutputs/C4_D007', ... (1 bytes))
0
Client (null) received PUBLISH (d0, q0, r0, m0, 'RESI_C4/DigitalOutputs/C4_D008', ... (1 bytes))
1
```



NodeRED® flow to read/write relay outputs to MQTT

If digital output changes send new state of all DOs to MQTT broker

The next flow will check cyclically, if the digital outputs have changed. If yes, a message for every digital output is send to the MQTT broker.



Edit function node 1

```
1 // create message with all DOs
2 var msg1={ topic: "C4_DOs", payload:[] };
3 var i
4
5 // n DOs
6 for (i = 0; i < global.get("C4_MAX_DOs"); i++) {
7   msg1.payload[i] = global.get("C4_DO" + String(i+1).padStart(2, '0'));
8 }
9
10 return msg1 ;
```

This node forms a message with an array of 8 elements. Each element in payload represents the actual state of a specific digital output.

Edit filter node 2

Mode: block unless value changes

Property: msg. payload

Apply mode separately for each

Property: msg. topic

Name: C4 DOs changed ?

This node waits until the message changes. This means unit one of the digital outputs has a new state

NodeRED® flow to read/write relay outputs to MQTT

Edit function node 3

Delete Cancel Done

Properties

Name C4 Create MQTT DOx data

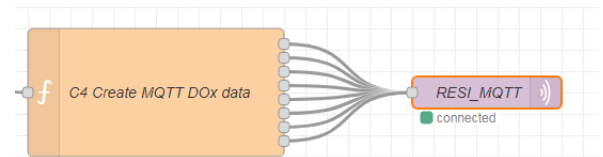
Setup On Start **On Message** On Stop

```
1 var msg1={
2   topic:"RESI_C4/DigitalOutputs/C4_DO01",
3   payload: msg.payload[0]
4 }
5
6 var msg2={
7   topic:"RESI_C4/DigitalOutputs/C4_DO02",
8   payload: msg.payload[1]
9 }
10
11 var msg3={
12   topic:"RESI_C4/DigitalOutputs/C4_DO03",
13   payload: msg.payload[2]
14 }
15
16 var msg4={
17   topic:"RESI_C4/DigitalOutputs/C4_DO04",
18   payload: msg.payload[3]
19 }
20
21 var msg5={
22   topic:"RESI_C4/DigitalOutputs/C4_DO05",
23   payload: msg.payload[4]
24 }
25
26 var msg6={
27   topic:"RESI_C4/DigitalOutputs/C4_DO06",
28   payload: msg.payload[5]
29 }
30
31 var msg7={
32   topic:"RESI_C4/DigitalOutputs/C4_DO07",
33   payload: msg.payload[6]
34 }
35
36 var msg8={
37   topic:"RESI_C4/DigitalOutputs/C4_DO08",
38   payload: msg.payload[7]
39 }
40
41
42 return [ msg1, msg2, msg3, msg4, msg5, msg6, msg7, msg8 ];
```

This node prepares 8 messages for the 8 digital outputs.

Every message topic is the MQTT reference for the output and the payload represents the actual state of the digital output: 0 for OFF and 1 for ON.

At the end of the function we return an array of eight messages for the eight output knots.



We connect all eight output knots to the same MQTT out node. This will trigger the send of eight individual MQTT messages.

Edit mqtt out node 4

Delete Cancel Done

Properties

Server 127.0.0.1:1883

Topic Topic

QoS 0 Retain true

Name RESI_MQTT

Tip: Leave topic, qos or retain blank if you want to set them via msg properties.



NodeRED® flow to build USER INTERFACE for DIP switch and LEDs

Read/Write the relay outputs from RESI-C4 controller to MQTT

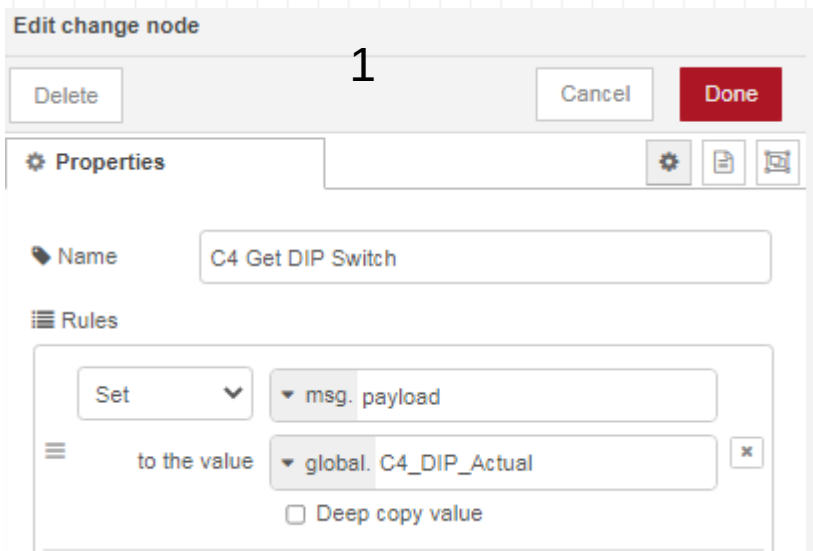
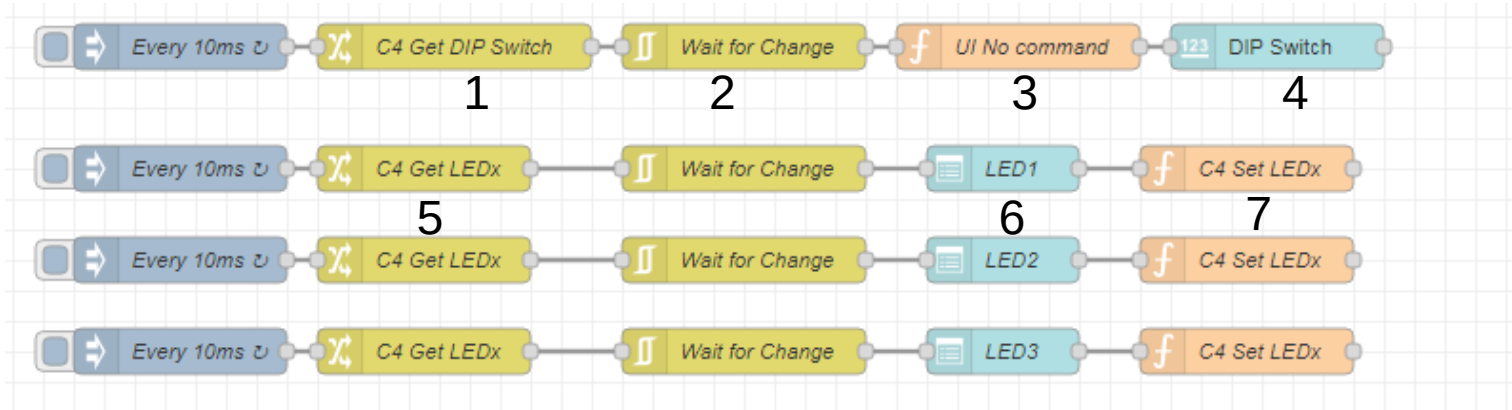
Now we create a new flow C4 MQTT. Import this flow:

```
[{"id":"d1b1b3188c87dd43","type":"tab","label":"C4 UI DIP+LEDs","disabled":false,"info":"","env":{}}, {"id":"7d39c75b5c25485e","type":"inject","z":"d1b1b3188c87dd43","name":"Every 10ms","props":[{"p":"payload"}], {"p":"topic","vt":"str"},"repeat":0.01,"crontab":"","once":false,"onceDelay":0.1,"topic":"","payloadType":"date","x":130,"y":40,"wires": [{"t":"set","p":"payload","pt":"msg","to":"C4_DIP_Actual","tot":"global"}], {"action":"","property":"","from":"","to":"","reg":false,"x":310,"y":40,"wires": [{"0baab220613ae508"}], {"id":"0baab220613ae508","type":"rbe","z":"d1b1b3188c87dd43","name":"Wait for Change","func":"rbe","gap":"","start":"","inout":"out","septomics":true,"property":"payload","topi":"topic","x":500,"y":40,"wires":[{"df2c62call26eef7"}], {"id":"144e0cf4e92f63b0","type":"inject","z":"d1b1b3188c87dd43","name":"Every 10ms","props":[{"p":"payload"}], {"p":"topic","vt":"str"},"repeat":0.01,"crontab":"","once":false,"onceDelay":0.1,"topic":"","payloadType":"date","x":130,"y":120,"wires": [{"2864439aaffacda5"}], {"id":"2864439aaffacda5","type":"change","z":"d1b1b3188c87dd43","name":"C4 Get LEDx","rules": [{"t":"set","p":"payload","pt":"msg","to":"C4_LED1_Actual","tot":"global"}], {"action":"","property":"","from":"","to":"","reg":false,"x":290,"y":120,"wires": [{"87c2df74c11b1e8f"}], {"id":"87c2df74c11b1e8f","type":"rbe","z":"d1b1b3188c87dd43","name":"Wait for Change","func":"rbe","gap":"","start":"","inout":"out","septomics":true,"property":"payload","topi":"topic","x":500,"y":120,"wires":[{"f40d944e15e9dada"}], {"id":"0a95cdd26afec0e8","type":"function","z":"d1b1b3188c87dd43","name":"C4 Set LEDx","func":"if (msg.payload >= 0 && msg.payload <= 5) {\n global.set(\"C4_LED1_State\", msg.payload);\n}\n\nreturn msg;","outputs":1,"timeout":0,"noerr":0,"initialize":"","finalize":"","libs": [{"x":830,"y":120,"wires":[]}], {"id":"0913752c44707eb5","type":"inject","z":"d1b1b3188c87dd43","name":"Every 10ms","props":[{"p":"payload"}], {"p":"topic","vt":"str"},"repeat":0.01,"crontab":"","once":false,"onceDelay":0.1,"topic":"","payloadType":"date","x":130,"y":180,"wires": [{"641c1d6947a563ff"}], {"id":"641c1d6947a563ff","type":"change","z":"d1b1b3188c87dd43","name":"C4 Get LEDx","rules": [{"t":"set","p":"payload","pt":"msg","to":"C4_LED2_Actual","tot":"global"}], {"action":"","property":"","from":"","to":"","reg":false,"x":290,"y":180,"wires": [{"663cef1bf1f3eb76"}], {"id":"663cef1bf1f3eb76","type":"rbe","z":"d1b1b3188c87dd43","name":"Wait for Change","func":"rbe","gap":"","start":"","inout":"out","septomics":true,"property":"payload","topi":"topic","x":500,"y":180,"wires":[{"919f9ac1225ab7cd"}], {"id":"f7d921c503210256","type":"function","z":"d1b1b3188c87dd43","name":"C4 Set LEDx","func":"if (msg.payload >= 0 && msg.payload <= 5) {\n global.set(\"C4_LED2_State\", msg.payload);\n}\n\nreturn msg;","outputs":1,"timeout":0,"noerr":0,"initialize":"","finalize":"","libs": [{"x":830,"y":180,"wires":[]}], {"id":"ef6e19959bd3f38d","type":"inject","z":"d1b1b3188c87dd43","name":"Every 10ms","props":[{"p":"payload"}], {"p":"topic","vt":"str"},"repeat":0.01,"crontab":"","once":false,"onceDelay":0.1,"topic":"","payloadType":"date","x":130,"y":240,"wires": [{"651b3a6698800270"}], {"id":"651b3a6698800270","type":"change","z":"d1b1b3188c87dd43","name":"C4 Get LEDx","rules": [{"t":"set","p":"payload","pt":"msg","to":"C4_LED3_Actual","tot":"global"}], {"action":"","property":"","from":"","to":"","reg":false,"x":290,"y":240,"wires": [{"a6f17620d260185f"}], {"id":"a6f17620d260185f","type":"rbe","z":"d1b1b3188c87dd43","name":"Wait for Change","func":"rbe","gap":"","start":"","inout":"out","septomics":true,"property":"payload","topi":"topic","x":500,"y":240,"wires":[{"1a390615587dc80e"}], {"id":"01114f6e6fa4be51","type":"function","z":"d1b1b3188c87dd43","name":"C4 Set LEDx","func":"if (msg.payload >= 0 && msg.payload <= 5) {\n global.set(\"C4_LED3_State\", msg.payload);\n}\n\nreturn msg;","outputs":1,"timeout":0,"noerr":0,"initialize":"","finalize":"","libs": [{"x":830,"y":240,"wires":[]}], {"id":"df2c62call26eef7","type":"function","z":"d1b1b3188c87dd43","name":"UI No command","func":"msg.enabled=false;\nreturn msg;","outputs":1,"timeout":0,"noerr":0,"initialize":"","finalize":"","libs": [{"x":680,"y":40,"wires": [{"46ab10f272e16bc4"}] }, {"id":"f40d944e15e9dada","type":"ui_dropdown","z":"d1b1b3188c87dd43","name":"LED1","label":"LED1","tooltip":"","place":"Select option","group":"961c9a07323de1db","order":2,"width":0,"height":0,"passthru":true,"multiple":false,"options":[{"label":"OFF","value":0,"type":"num"}, {"label":"ON","value":1,"type":"num"}, {"label":"INVER","value":2,"type":"num"}, {"label":"BLINK","value":3,"type":"num"}, {"label":"FLASH","value":4,"type":"num"}], {"label":"PULSE","value":5,"type":"num"}], {"payload":"","topic":"topic","topicType":"msg","className":"","x":670,"y":120,"wires":[{"0a95cdd26afec0e8"}], {"id":"919f9ac1225ab7cd","type":"ui_dropdown","z":"d1b1b3188c87dd43","name":"LED2","label":"LED2","tooltip":"","place":"Select option","group":"961c9a07323de1db","order":3,"width":0,"height":0,"passthru":true,"multiple":false,"options":[{"label":"OFF","value":0,"type":"num"}, {"label":"ON","value":1,"type":"num"}, {"label":"INVER","value":2,"type":"num"}, {"label":"BLINK","value":3,"type":"num"}, {"label":"FLASH","value":4,"type":"num"}], {"label":"PULSE","value":5,"type":"num"}], {"payload":"","topic":"topic","topicType":"msg","className":"","x":670,"y":180,"wires":[{"f7d921c503210256"}], {"id":"1a390615587dc80e","type":"ui_dropdown","z":"d1b1b3188c87dd43","name":"LED3","label":"LED3","tooltip":"","place":"Select option","group":"961c9a07323de1db","order":4,"width":0,"height":0,"passthru":true,"multiple":false,"options":[{"label":"OFF","value":0,"type":"num"}, {"label":"ON","value":1,"type":"num"}, {"label":"INVER","value":2,"type":"num"}, {"label":"BLINK","value":3,"type":"num"}, {"label":"FLASH","value":4,"type":"num"}], {"label":"PULSE","value":5,"type":"num"}], {"payload":"","topic":"topic","topicType":"msg","className":"","x":670,"y":240,"wires":[{"01114f6e6fa4be51"}], {"id":"46ab10f272e16bc4","type":"ui_numeric","z":"d1b1b3188c87dd43","name":"","label":"DIP Switch","tooltip":"","group":"2afc92d9ce59f282","order":1,"width":0,"height":0,"wrap":false,"passthru":true,"topic":"topic","topicType":"msg","format":"{value}","min":0,"max":255,"step":1,"className":"","x":850,"y":40,"wires":[]}], {"id":"961c9a07323de1db","type":"ui_group","name":"LEDs","tab":"ce994ee19fec0ede","order":2,"disp":true,"width":6,"collapse":false,"className":""}, {"id":"2afc92d9ce59f282","type":"ui_group","name":"DIP Switch","tab":"ce994ee19fec0ede","order":1,"disp":true,"width":6,"collapse":false,"className":""}, {"id":"ce994ee19fec0ede","type":"ui_tab","name":"C4 DIP+LEDs","icon":"dashboard","order":2,"disabled":false,"hidden":false}]
```

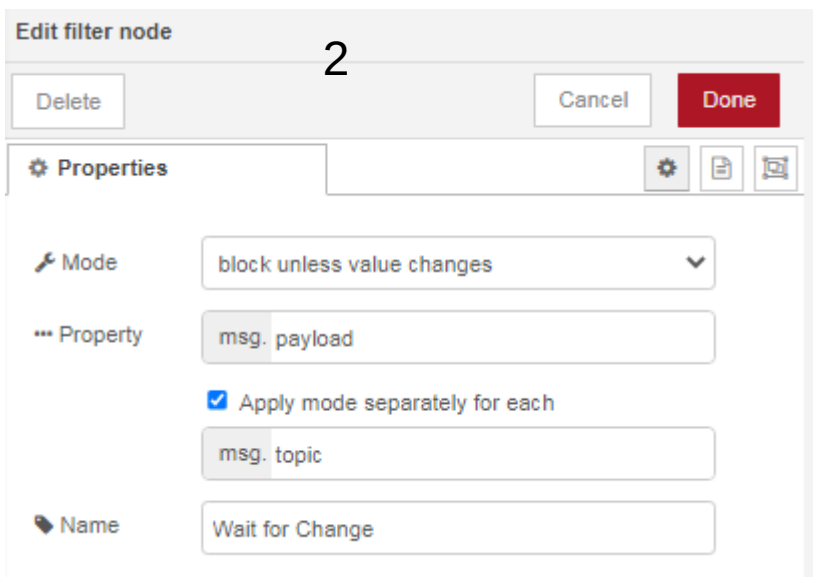
NodeRED® flow to build USER INTERFACE for DIP switch and LEDs

Create flow for UI DIP+LEDs

We create a new flow with the title C4 UI DIP+LEDs.



This node copies the contents of the global variable C4_DIP_Actual to the message for the next node



This node waits until the message changes. This means the next node is activated only, if the DIP switch has changed.



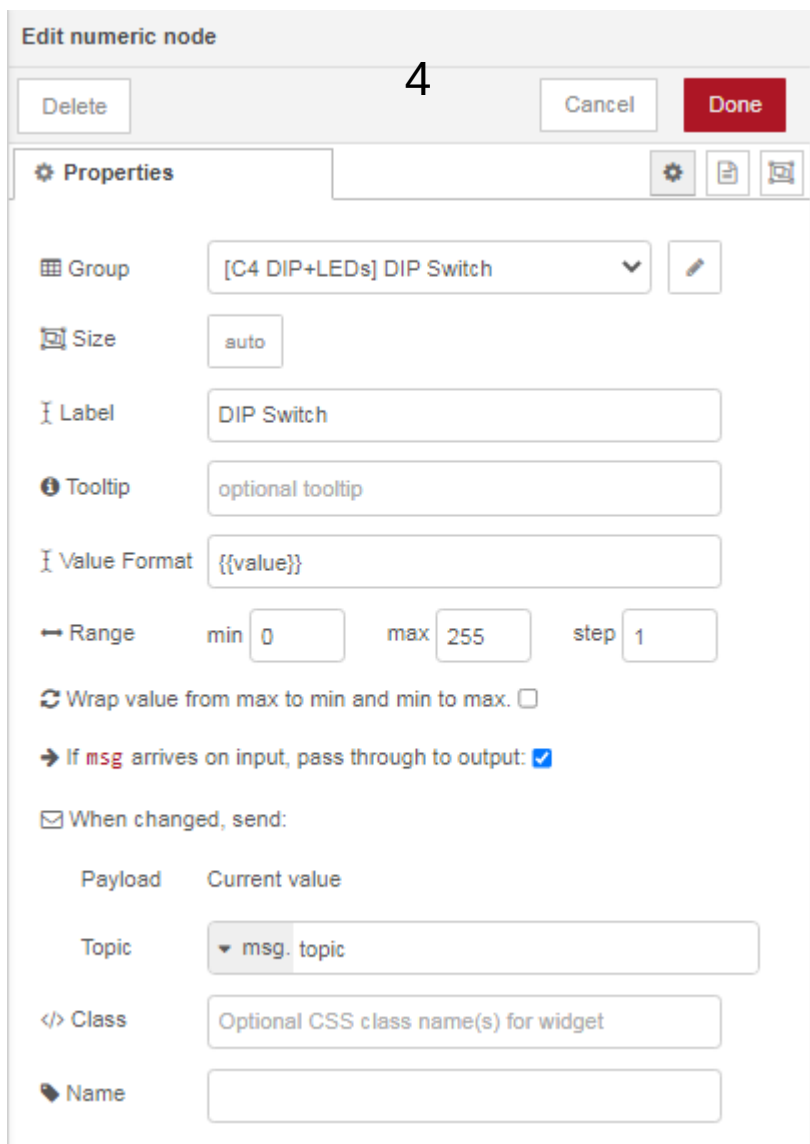
NodeRED® flow to build USER INTERFACE for DIP switch and LEDs



With this function node, we instruct the next node, which is the UI node, that the control is only for viewing, but not for control.

Therefore we set `msg.enabled` to false.

The payload of the message is the current value of the DIP switch.



We choose the numeric node to display the current value of the DIP switch.

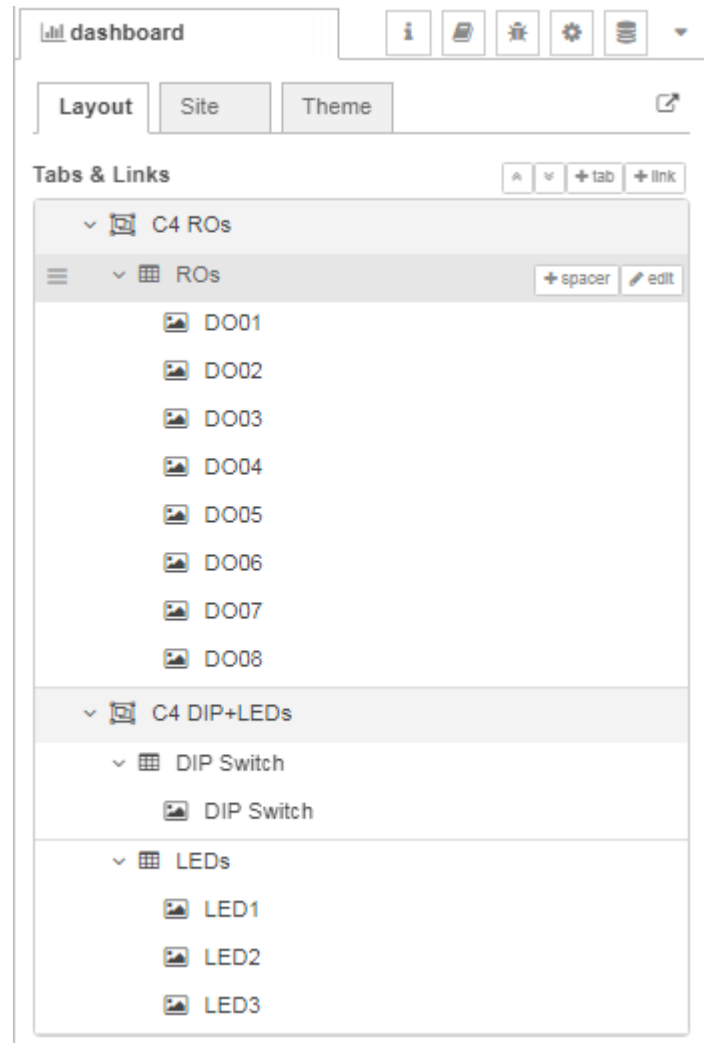
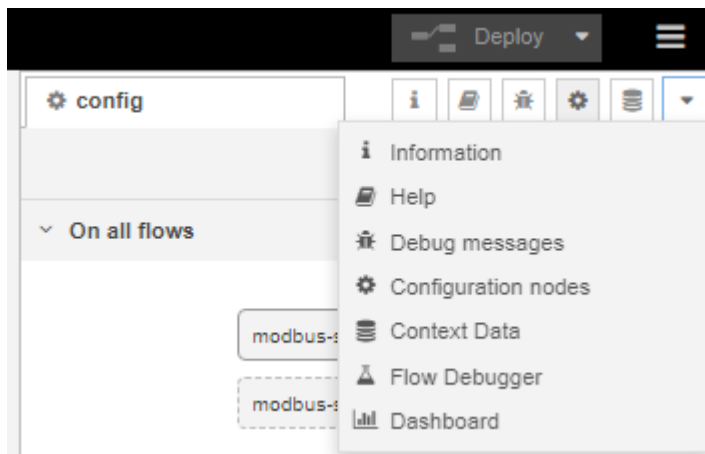


NodeRED® flow to build USER INTERFACE for DIP switch and LEDs

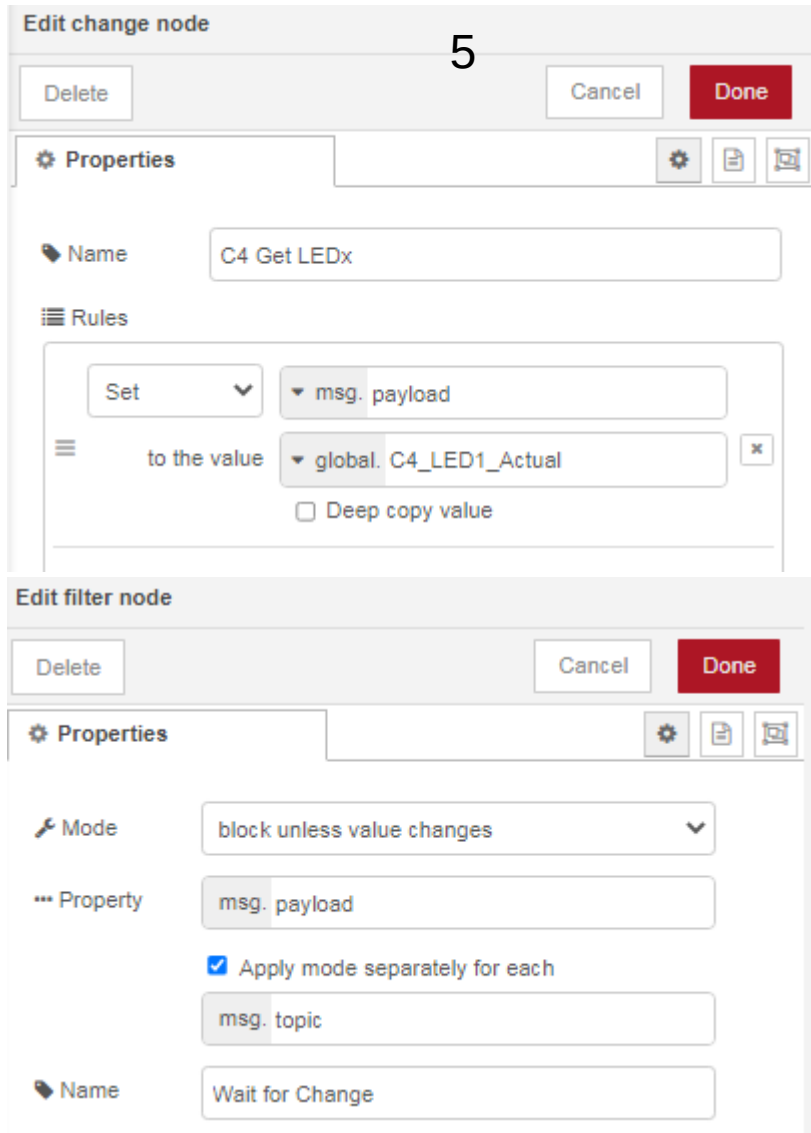
HOWTO build simple dashboard for DIP switches and LEDs

First you have to install the component `node-red-dashboard`.

Now open the new menu Dashboard. Create the tabs C4 ROs and C4 DIP+LEDs. Within the tab C4 ROs create the group node ROs. Within the tab C3 DIP+LEDs create the group node DIP Switch and the group node LEDs



NodeRED® flow to build USER INTERFACE for DIP switch and LEDs



We copy with this node the contents of the global variable C4_LED1-Actual to the message payload.

Then we wait for an change in the value to update the UI node



NodeRED® flow to build USER INTERFACE for DIP switch and LEDs



We choose a drop down UI interface element to show the actual status of the LED and to select a new command for the LED. The other two LEDs operate in the same way.



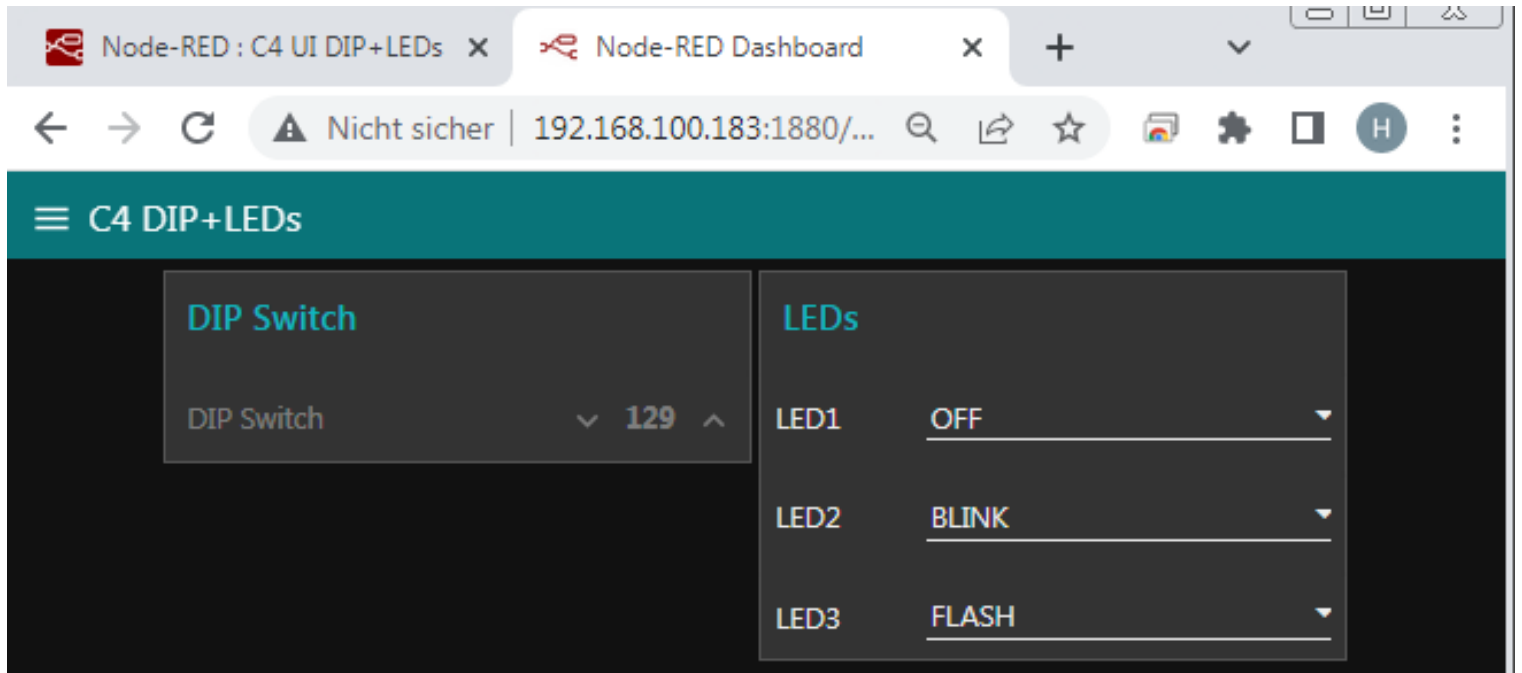
When the user selects a new mode from the drop down list, this node will check, if the value is correct and updates the global variable C4_LED1_State.

The flow C4 DIP Switch+LEDs will react on the change and write via MODBUS the new mode value to the affected LED.



NodeRED® flow to build USER INTERFACE for DIP switch and LEDs

Open your browser and enter the correct URL for your UI. You can now select a new mode for the three LEDs and if you change the DIP switch, the shown value will change too. Select from the Drop Down Menu the correct page.



NodeRED® flow to build USER INTERFACE for relay outputs

Read/Write the relay outputs from RESI-C4 controller to MQTT

Now we create a new flow C4 MQTT. Import this flow:

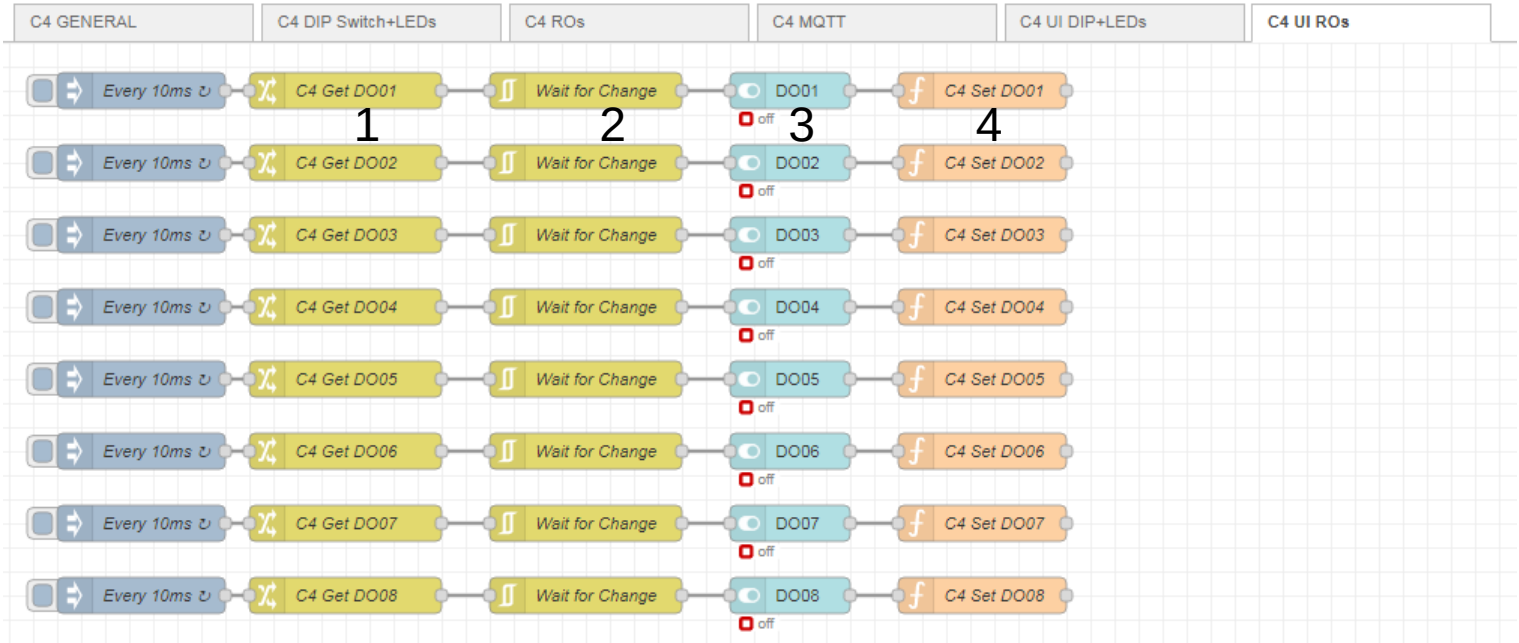
```
[{"id":"eb35aa3e56e41af4","type":"tab","label":"C4 UI ROs","disabled":false,"info":"","env":{}}, {"id":"553210f3700e2164","type":"inject","z":"eb35aa3e56e41af4","name":"Every 10ms","props":[{"p":"payload"}], {"p":"topic","vt":"str"},"repeat":"0.01","crontab":"*","once":false,"onceDelay":0.1,"topic":"","payload":"","payloadType":"date","x":130,"y":40,"wires": [{"980a4fb303e6e960"}]}, {"id":"980a4fb303e6e960","type":"change","z":"eb35aa3e56e41af4","name":"C4 Get DO01","rules": [{"t":"set","p":"payload","pt":"msg","to":"C4_DO01","tot":"global"}],"action":"","property":"","from":"","to":"","reg":false,"x":300,"y":40,"wires": [{"b0519ec6474e960c"}]}, {"id":"b0519ec6474e960c","type":"rbe","z":"eb35aa3e56e41af4","name":"Wait for Change","func":"rbe","gap":"","start":"","inout":"out","septomics":true,"property":"payload","topic":"topic","x":500,"y":40,"wires": [{"33848eb8ac54d278"}]}, {"id":"33848eb8ac54d278","type":"inject","z":"eb35aa3e56e41af4","name":"Every 10ms","props":[{"p":"payload"}], {"p":"topic","vt":"str"},"repeat":"0.01","crontab":"*","once":false,"onceDelay":0.1,"topic":"","payload":"","payloadType":"date","x":130,"y":100,"wires": [{"23d16de9278e3d5e"}]}, {"id":"23d16de9278e3d5e","type":"change","z":"eb35aa3e56e41af4","name":"C4 Get DO02","rules": [{"t":"set","p":"payload","pt":"msg","to":"C4_DO02","tot":"global"}],"action":"","property":"","from":"","to":"","reg":false,"x":300,"y":100,"wires": [{"3b94740763921fb8"}]}, {"id":"3b94740763921fb8","type":"rbe","z":"eb35aa3e56e41af4","name":"Wait for Change","func":"rbe","gap":"","start":"","inout":"out","septomics":true,"property":"payload","topic":"topic","x":500,"y":100,"wires": [{"035bf1e7876cb84"}]}, {"id":"035bf1e7876cb84","type":"inject","z":"eb35aa3e56e41af4","name":"Every 10ms","props":[{"p":"payload"}], {"p":"topic","vt":"str"},"repeat":"0.01","crontab":"*","once":false,"onceDelay":0.1,"topic":"","payload":"","payloadType":"date","x":130,"y":160,"wires": [{"42316f7981b5fe8c"}]}, {"id":"42316f7981b5fe8c","type":"change","z":"eb35aa3e56e41af4","name":"C4 Get DO03","rules": [{"t":"set","p":"payload","pt":"msg","to":"C4_DO03","tot":"global"}],"action":"","property":"","from":"","to":"","reg":false,"x":300,"y":160,"wires": [{"180fde5ad067cd09"}]}, {"id":"180fde5ad067cd09","type":"rbe","z":"eb35aa3e56e41af4","name":"Wait for Change","func":"rbe","gap":"","start":"","inout":"out","septomics":true,"property":"payload","topic":"topic","x":500,"y":160,"wires": [{"1e51f53d2bc620fe"}]}, {"id":"1e51f53d2bc620fe","type":"inject","z":"eb35aa3e56e41af4","name":"Every 10ms","props":[{"p":"payload"}], {"p":"topic","vt":"str"},"repeat":"0.01","crontab":"*","once":false,"onceDelay":0.1,"topic":"","payload":"","payloadType":"date","x":130,"y":220,"wires": [{"56d2b13f3ca96272"}]}, {"id":"56d2b13f3ca96272","type":"change","z":"eb35aa3e56e41af4","name":"C4 Get DO04","rules": [{"t":"set","p":"payload","pt":"msg","to":"C4_DO04","tot":"global"}],"action":"","property":"","from":"","to":"","reg":false,"x":300,"y":220,"wires": [{"6d6be841ccf1a3de"}]}, {"id":"6d6be841ccf1a3de","type":"rbe","z":"eb35aa3e56e41af4","name":"Wait for Change","func":"rbe","gap":"","start":"","inout":"out","septomics":true,"property":"payload","topic":"topic","x":500,"y":220,"wires": [{"00e0e064cb05559"}]}, {"id":"00e0e064cb05559","type":"inject","z":"eb35aa3e56e41af4","name":"Every 10ms","props":[{"p":"payload"}], {"p":"topic","vt":"str"},"repeat":"0.01","crontab":"*","once":false,"onceDelay":0.1,"topic":"","payload":"","payloadType":"date","x":130,"y":280,"wires": [{"489a9820c81e935a"}]}, {"id":"489a9820c81e935a","type":"change","z":"eb35aa3e56e41af4","name":"C4 Get DO05","rules": [{"t":"set","p":"payload","pt":"msg","to":"C4_DO05","tot":"global"}],"action":"","property":"","from":"","to":"","reg":false,"x":300,"y":280,"wires": [{"bb4dacl1bb4d69533"}]}, {"id":"bb4dacl1bb4d69533","type":"rbe","z":"eb35aa3e56e41af4","name":"Wait for Change","func":"rbe","gap":"","start":"","inout":"out","septomics":true,"property":"payload","topic":"topic","x":500,"y":280,"wires": [{"4922771ce08f80c1"}]}, {"id":"4922771ce08f80c1","type":"inject","z":"eb35aa3e56e41af4","name":"Every 10ms","props":[{"p":"payload"}], {"p":"topic","vt":"str"},"repeat":"0.01","crontab":"*","once":false,"onceDelay":0.1,"topic":"","payload":"","payloadType":"date","x":130,"y":340,"wires": [{"9eb44838e9f3df87"}]}, {"id":"9eb44838e9f3df87","type":"change","z":"eb35aa3e56e41af4","name":"C4 Get DO06","rules": [{"t":"set","p":"payload","pt":"msg","to":"C4_DO06","tot":"global"}],"action":"","property":"","from":"","to":"","reg":false,"x":300,"y":340,"wires": [{"e479c5579d3071a3"}]}, {"id":"e479c5579d3071a3","type":"rbe","z":"eb35aa3e56e41af4","name":"Wait for Change","func":"rbe","gap":"","start":"","inout":"out","septomics":true,"property":"payload","topic":"topic","x":500,"y":340,"wires": [{"07b7792b68246ac2"}]}, {"id":"07b7792b68246ac2","type":"inject","z":"eb35aa3e56e41af4","name":"Every 10ms","props":[{"p":"payload"}], {"p":"topic","vt":"str"},"repeat":"0.01","crontab":"*","once":false,"onceDelay":0.1,"topic":"","payload":"","payloadType":"date","x":130,"y":400,"wires": [{"67c7a85b5709b123"}]}, {"id":"67c7a85b5709b123","type":"change","z":"eb35aa3e56e41af4","name":"C4 Get DO07","rules": [{"t":"set","p":"payload","pt":"msg","to":"C4_DO07","tot":"global"}],"action":"","property":"","from":"","to":"","reg":false,"x":300,"y":400,"wires": [{"eebf5134f460b624"}]}, {"id":"eebf5134f460b624","type":"rbe","z":"eb35aa3e56e41af4","name":"Wait for Change","func":"rbe","gap":"","start":"","inout":"out","septomics":true,"property":"payload","topic":"topic","x":500,"y":400,"wires": [{"a200f1f3f9cd025c"}]}, {"id":"a200f1f3f9cd025c","type":"inject","z":"eb35aa3e56e41af4","name":"Every 10ms","props":[{"p":"payload"}], {"p":"topic","vt":"str"},"repeat":"0.01","crontab":"*","once":false,"onceDelay":0.1,"topic":"","payload":"","payloadType":"date","x":130,"y":460,"wires": [{"44c7ae4c7913eab27"}]}, {"id":"44c7ae4c7913eab27","type":"change","z":"eb35aa3e56e41af4","name":"C4 Get DO08","rules": [{"t":"set","p":"payload","pt":"msg","to":"C4_DO08","tot":"global"}],"action":"","property":"","from":"","to":"","reg":false,"x":300,"y":460,"wires": [{"ac6929c7ec529620"}]}, {"id":"ac6929c7ec529620","type":"rbe","z":"eb35aa3e56e41af4","name":"Wait for Change","func":"rbe","gap":"","start":"","inout":"out","septomics":true,"property":"payload","topic":"topic","x":500,"y":460,"wires": [{"5a154b4fb5c4f03"}]}, {"id":"5a154b4fb5c4f03","type":"function","z":"eb35aa3e56e41af4","name":"C4 Set DO01","func":"if (msg.payload == 0 || msg.payload == 1) {\n global.set(\"C4_DO01\", msg.payload);\n}\n\nreturn msg;","outputs":1,"timeout":0,"noerr":0,"initialize":"","finalize":"","libs":[],"x":830,"y":40,"wires": [{"e41ecc0b36f88b7c"}]}, {"id":"e41ecc0b36f88b7c","type":"function","z":"eb35aa3e56e41af4","name":"C4 Set DO02","func":"if (msg.payload == 0 || msg.payload == 1) {\n global.set(\"C4_DO02\", msg.payload);\n}\n\nreturn msg;","outputs":1,"timeout":0,"noerr":0,"initialize":"","finalize":"","libs":[],"x":830,"y":100,"wires": [{"a81f554348694777"}]}, {"id":"a81f554348694777","type":"function","z":"eb35aa3e56e41af4","name":"C4 Set DO03","func":"if (msg.payload == 0 || msg.payload == 1) {\n global.set(\"C4_DO03\", msg.payload);\n}\n\nreturn msg;","outputs":1,"timeout":0,"noerr":0,"initialize":"","finalize":"","libs":[],"x":830,"y":160,"wires": [{"dcb18b4607e155b1"}]}, {"id":"dcb18b4607e155b1","type":"function","z":"eb35aa3e56e41af4","name":"C4 Set DO04","func":"if (msg.payload == 0 || msg.payload == 1) {\n global.set(\"C4_DO04\", msg.payload);\n}\n\nreturn msg;","outputs":1,"timeout":0,"noerr":0,"initialize":"","finalize":"","libs":[],"x":830,"y":220,"wires": [{"5d254b2a954b575d"}]}, {"id":"5d254b2a954b575d","type":"function","z":"eb35aa3e56e41af4","name":"C4 Set DO05","func":"if (msg.payload == 0 || msg.payload == 1) {\n global.set(\"C4_DO05\", msg.payload);\n}\n\nreturn msg;","outputs":1,"timeout":0,"noerr":0,"initialize":"","finalize":"","libs":[],"x":830,"y":280,"wires": [{"747fa053b63957c75"}]}, {"id":"747fa053b63957c75","type":"function","z":"eb35aa3e56e41af4","name":"C4 Set DO06","func":"if (msg.payload == 0 || msg.payload == 1) {\n global.set(\"C4_DO06\", msg.payload);\n}\n\nreturn msg;","outputs":1,"timeout":0,"noerr":0,"initialize":"","finalize":"","libs":[],"x":830,"y":340,"wires": [{"18163a82797f46b5"}]}, {"id":"18163a82797f46b5","type":"function","z":"eb35aa3e56e41af4","name":"C4 Set DO07","func":"if (msg.payload == 0 || msg.payload == 1) {\n global.set(\"C4_DO07\", msg.payload);\n}\n\nreturn msg;","outputs":1,"timeout":0,"noerr":0,"initialize":"","finalize":"","libs":[],"x":830,"y":400,"wires": [{"e0bb9dc76f9a20f0"}]}, {"id":"e0bb9dc76f9a20f0","type":"function","z":"eb35aa3e56e41af4","name":"C4 Set DO08","func":"if (msg.payload == 0 || msg.payload == 1) {\n global.set(\"C4_DO08\", msg.payload);\n}\n\nreturn msg;","outputs":1,"timeout":0,"noerr":0,"initialize":"","finalize":"","libs":[],"x":830,"y":460,"wires": [{"33848eb8ac54d278"}]}, {"id":"33848eb8ac54d278","type":"ui_switch","z":"eb35aa3e56e41af4","name":"","label":"DO01","tooltip":"","group":"6a3calb5503c168a","order":1,"width":0,"height":0,"passthru":true,"decouple":"false","topic":"topic","topicType":"msg","style":"","onvalue":"1","onvalueType":"num","onicon":"lightbulb_outline","oncolor":"yellow","offvalue":"0","offvalueType":"num","officon":"lightbulb_outline","offcolor":"gray","animate":true,"className":"","x":670,"y":40,"wires": [{"6a7473fee23a66f9"}]}, {"id":"6a7473fee23a66f9","type":"ui_switch","z":"eb35aa3e56e41af4","name":"","label":"DO02","tooltip":"","group":"6a3calb5503c168a","order":2,"width":0,"height":0,"passthru":true,"decouple":"false","topic":"topic","topicType":"msg","style":"","onvalue":"1","onvalueType":"num","onicon":"lightbulb_outline","oncolor":"yellow","offvalue":"0","offvalueType":"num","officon":"lightbulb_outline","offcolor":"gray","animate":true,"className":"","x":670,"y":100,"wires": [{"e41ecc0b36f88b7c"}]}, {"id":"e41ecc0b36f88b7c","type":"ui_switch","z":"eb35aa3e56e41af4","name":"","label":"DO03","tooltip":"","group":"6a3calb5503c168a","order":3,"width":0,"height":0,"passthru":true,"decouple":"false","topic":"topic","topicType":"msg","style":"","onvalue":"1","onvalueType":"num","onicon":"lightbulb_outline","oncolor":"yellow","offvalue":"0","offvalueType":"num","officon":"lightbulb_outline","offcolor":"gray","animate":true,"className":"","x":670,"y":160,"wires": [{"a81f554348694777"}]}, {"id":"a81f554348694777","type":"ui_switch","z":"eb35aa3e56e41af4","name":"","label":"DO04","tooltip":"","group":"6a3calb5503c168a","order":4,"width":0,"height":0,"passthru":true,"decouple":"false","topic":"topic","topicType":"msg","style":"","onvalue":"1","onvalueType":"num","onicon":"lightbulb_outline","oncolor":"yellow","offvalue":"0","offvalueType":"num","officon":"lightbulb_outline","offcolor":"gray","animate":true,"className":"","x":670,"y":220,"wires": [{"dcb18b4607e155b1"}]}, {"id":"dcb18b4607e155b1","type":"ui_switch","z":"eb35aa3e56e41af4","name":"","label":"DO05","tooltip":"","group":"6a3calb5503c168a","order":5,"width":0,"height":0,"passthru":true,"decouple":"false","topic":"topic","topicType":"msg","style":"","onvalue":"1","onvalueType":"num","onicon":"lightbulb_outline","oncolor":"yellow","offvalue":"0","offvalueType":"num","officon":"lightbulb_outline","offcolor":"gray","animate":true,"className":"","x":670,"y":280,"wires": [{"5d254b2a954b575d"}]}, {"id":"5d254b2a954b575d","type":"ui_switch","z":"eb35aa3e56e41af4","name":"","label":"DO06","tooltip":"","group":"6a3calb5503c168a","order":6,"width":0,"height":0,"passthru":true,"decouple":"false","topic":"topic","topicType":"msg","style":"","onvalue":"1","onvalueType":"num","onicon":"lightbulb_outline","oncolor":"yellow","offvalue":"0","offvalueType":"num","officon":"lightbulb_outline","offcolor":"gray","animate":true,"className":"","x":670,"y":340,"wires": [{"747fa053b63957c75"}]}, {"id":"747fa053b63957c75","type":"ui_switch","z":"eb35aa3e56e41af4","name":"","label":"DO07","tooltip":"","group":"6a3calb5503c168a","order":7,"width":0,"height":0,"passthru":true,"decouple":"false","topic":"topic","topicType":"msg","style":"","onvalue":"1","onvalueType":"num","onicon":"lightbulb_outline","oncolor":"yellow","offvalue":"0","offvalueType":"num","officon":"lightbulb_outline","offcolor":"gray","animate":true,"className":"","x":670,"y":400,"wires": [{"18163a82797f46b5"}]}, {"id":"18163a82797f46b5","type":"ui_switch","z":"eb35aa3e56e41af4","name":"","label":"DO08","tooltip":"","group":"6a3calb5503c168a","order":8,"width":0,"height":0,"passthru":true,"decouple":"false","topic":"topic","topicType":"msg","style":"","onvalue":"1","onvalueType":"num","onicon":"lightbulb_outline","oncolor":"yellow","offvalue":"0","offvalueType":"num","officon":"lightbulb_outline","offcolor":"gray","animate":true,"className":"","x":670,"y":460,"wires": [{"33848eb8ac54d278"}]}]
```

AN-46 More information under www.raspberrypi.org  

NodeRED® flow to build USER INTERFACE for relay outputs

Create flow for UI ROs

We create a new flow with the title C4 UI ROs



Edit change node

1

Delete Cancel Done

Properties

Name: C4 Get DO01

Rules

Set msg. payload to the value global. C4_DO01

Deep copy value

This node copies the contents of the global variable C4_DO01 to the message for the next node

Edit filter node

Delete Cancel Done

Properties

Mode: block unless value changes

Property: msg. payload

Apply mode separately for each

msg. topic

Name: Wait for Change

This node waits until the message changes. This means the next node is activated only, if the DIP switch has changed.



NodeRED® flow to build USER INTERFACE for relay outputs

3

Delete Cancel Done

Properties

Group [C4 ROs] ROs

Size auto

Label DO01

Tooltip optional tooltip

Icon Custom 4 Animate

On Icon lightbulb_outline Colour yellow

Off Icon lightbulb_outline Colour gray

Pass through msg if payload matches valid state:

When clicked, send:

On Payload 1

Off Payload 0

Topic msg.topic

Class Optional CSS class name(s) for widget

Name

With this UI node we display a switch. We use a light bulb as symbol. It will be yellow, if DO is ON and gray if DO is OFF.

Also we allow switching the state to 0 or 1.

4

Delete Cancel Done

Properties

Name C4 Set DO01

Setup On Start On Message On Stop

```
1 if (msg.payload == 0 || msg.payload == 1) {
2   global.set("C4_DO01", msg.payload);
3 }
4
5 return msg;
```

When the user clicks onto the light bulb, this node will be activated with the new state (0 or 1).

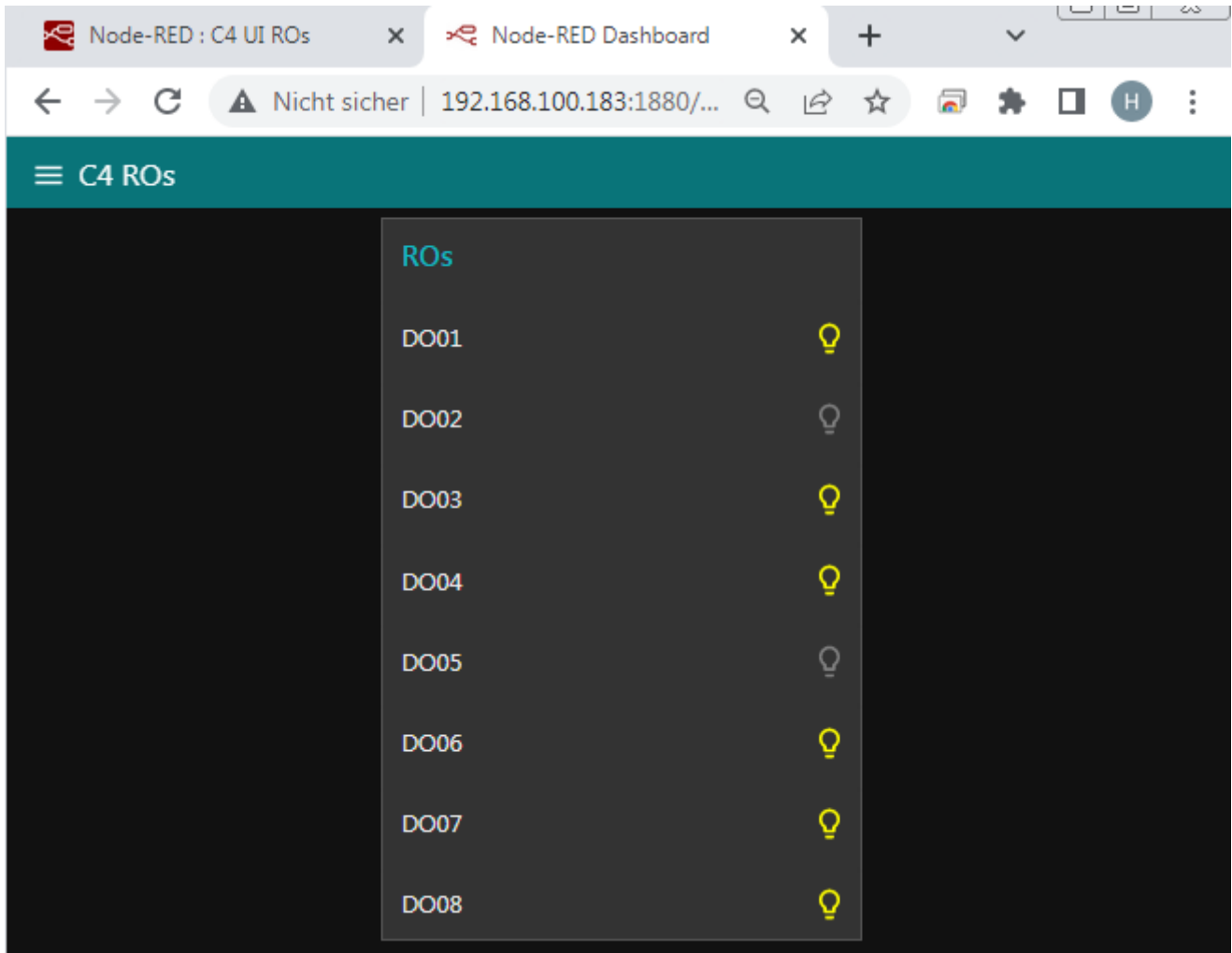
This node sets the corresponding global variable C4_DOxx to the new value.

The flow C4 ROs will then update the digital outputs via MODBUS.



NodeRED® flow to build USER INTERFACE for relay outputs

Open your browser and enter the correct URL for your UI. You can now switch the digital outputs by clicking on the light bulb. But if you send a MQTT message to switch the digital output the new status will be shown also. Select from the Drop Down Menu the correct page.



RESI Informatik & Automation GmbH
Altenmarkt 29, A-8551 Wies, AUSTRIA
help@RESI.cc www.RESI.cc

