How to use the PFC-200 serial port with Node-RED

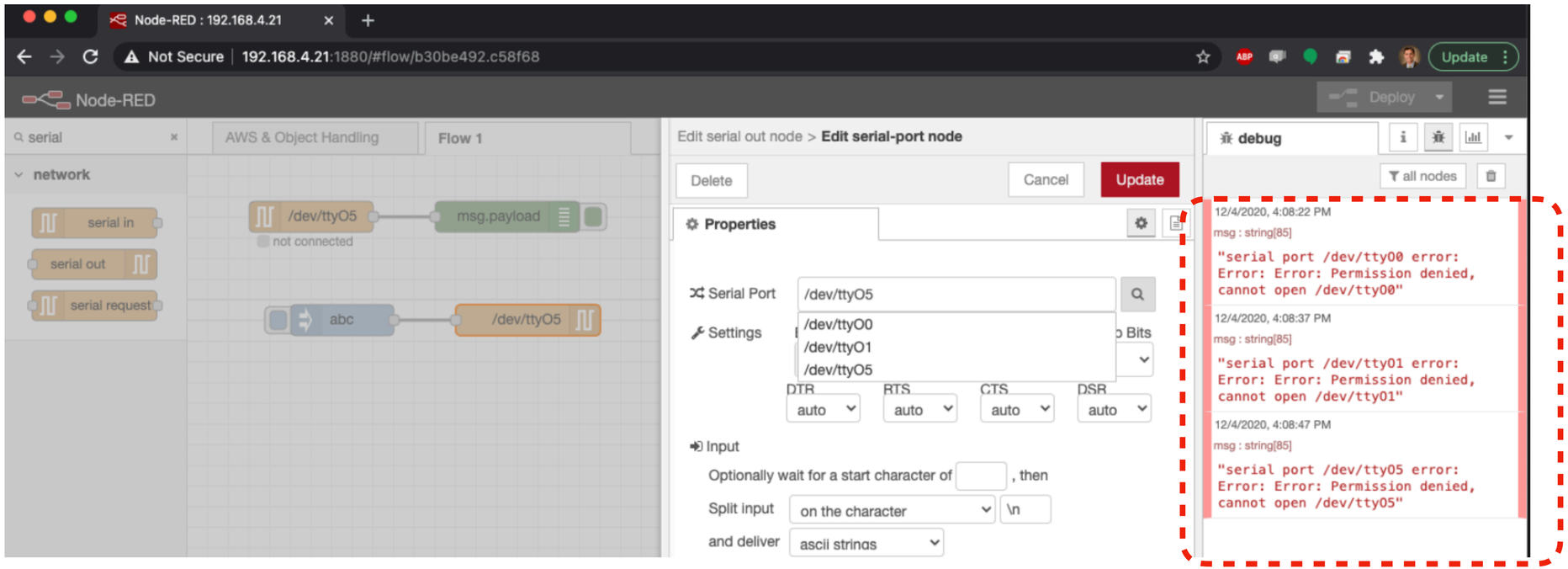
Kurt Braun
December 2020

Question: Why do I get a permission denied error when I try to use the onboard serial port in a Node-RED container?

In order to use the onboard serial port of the PFC200 in Node-RED some configuration changes must be made, and the container must have the device attached to it.

This procedure will explain the steps to get the serial port to work in Node-RED.

```
root@PFC200V3-46E709:/dev docker run --rm -t --network=host -v node_red_user_data:/data --device=/dev/ttyO1:/dev/ttyO1:rw nodered/node-red
11 Dec 12:11:56 - [info] Started flows
11 Dec 12:11:56 - [error] serial port /dev/ttyO1 error: Error: Error: Permission denied, cannot open /dev/ttyO1
```



Permission denied error in Node-RED

Question: How can I identify the onboard serial port /dev/ name?

The linux command `ls /dev -l` shows all the serial ports found on the controller (partial list shown below for clarity).

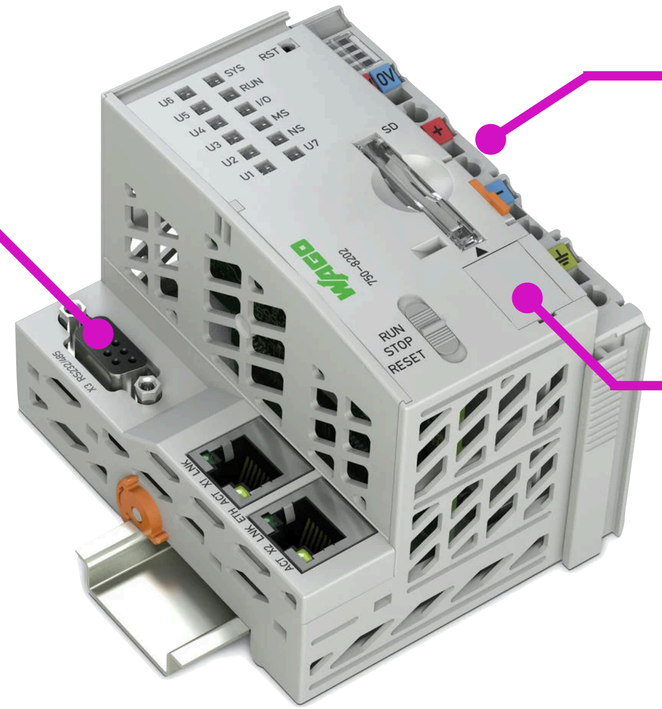
```
root@PFC200V3-46E709:/dev ls -l
c---------    1 root     root       10, 235 Dec 11 02:48 autofs
lrwxrwxrwx    1 root     root             10 Dec 11 02:48 serial -> /dev/ttyO1
lrwxrwxrwx    1 root     root             10 Dec 11 02:48 service -> /dev/ttyO0
lrwxrwxrwx    1 root     root             10 Dec 11 02:48 ttyKbus -> /dev/ttyO5
crw-rw----    1 root     dialout   247,    0 Dec 11 02:48 ttyO0
crw-rw----    1 root     dialout   246,    1 Dec 11 02:48 ttyO1
crw-rw----    1 root     dialout   247,    5 Dec 11 02:48 ttyO5
```

Soft Links (blue) are used help you easily identifiable the tty serial ports.

The on-board serial port of the PFC200 is /dev/ttyO1.

Serial
/dev/ttyO1

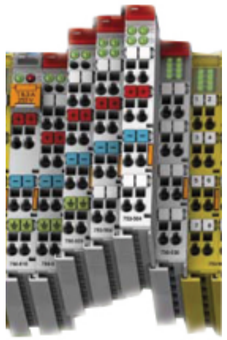
Note: This is O as in Omega, not the number 0.



KBUS
/dev/ttyO5
Do not use

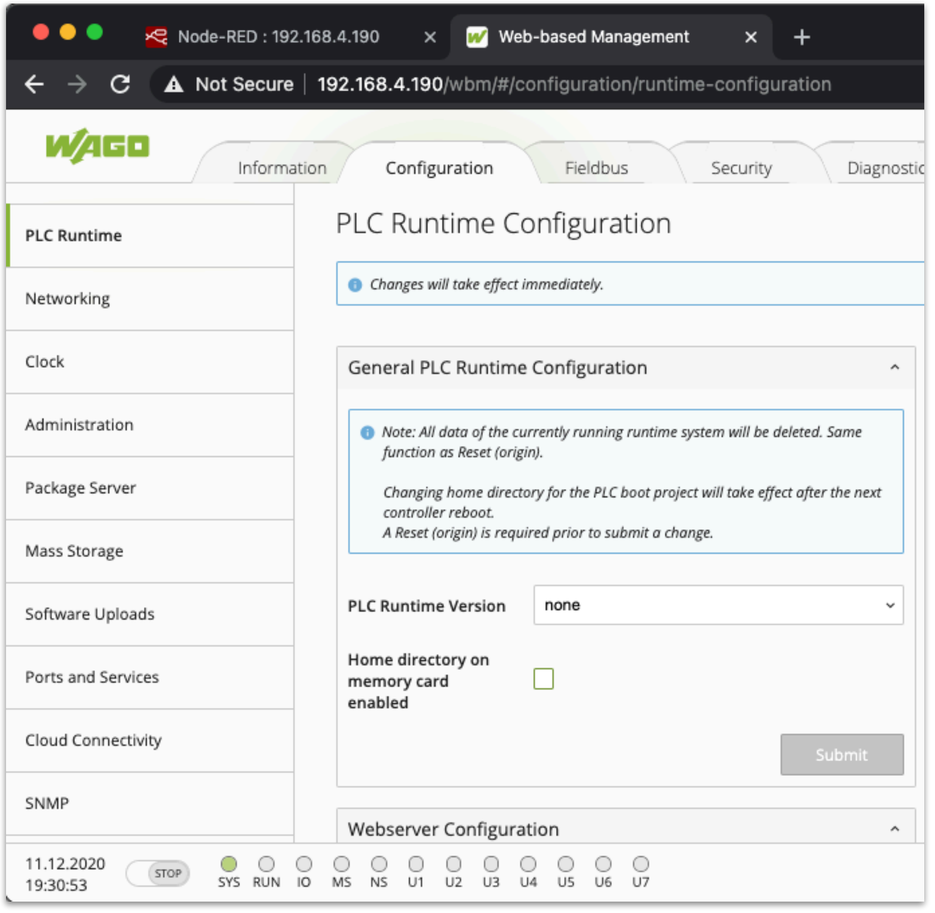
Service
/dev/ttyO0
Do not use

I/O Modules

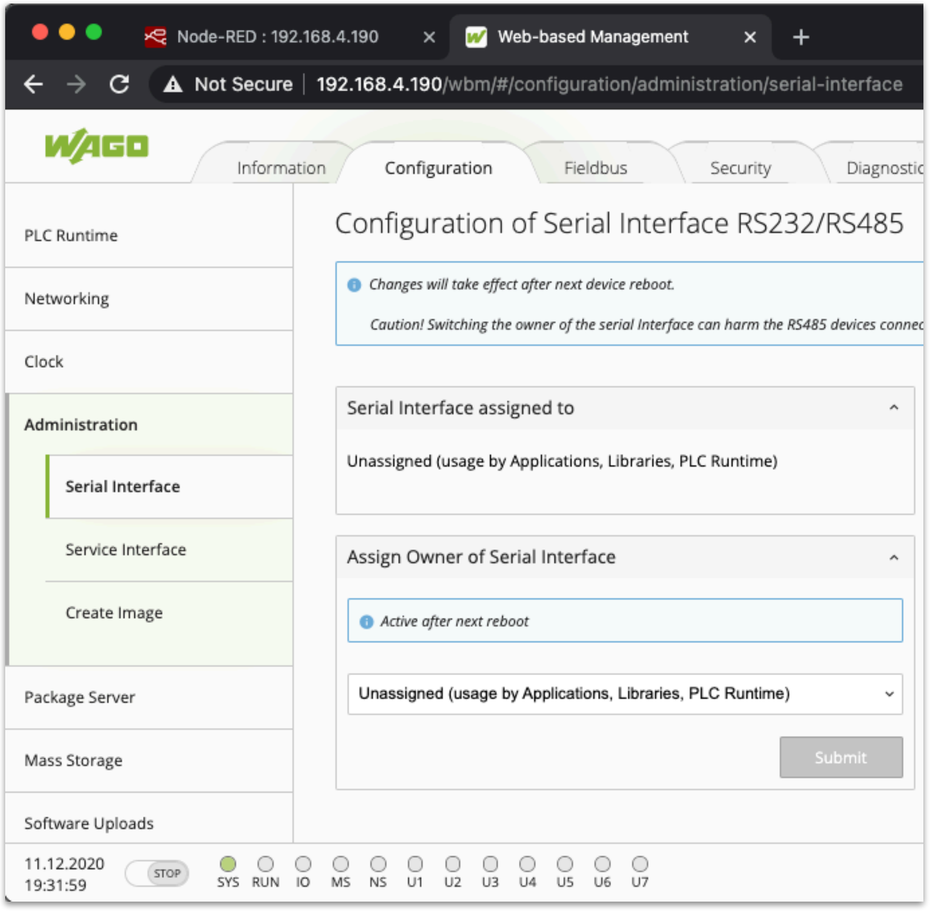


Step 1:

Set the runtime to none in the Web Based Management.



Set Serial Interface to “Unnassigned”



Step 2:

The serial port device has insufficient permissions to use with Docker. This can be checked by looking at the results of `ls -l` command.

```
root@PFC200V3-46E709:/dev ls -l
crw-rw----    1 root     dialout   246,   1 Dec 11 02:48 ttyO1
```

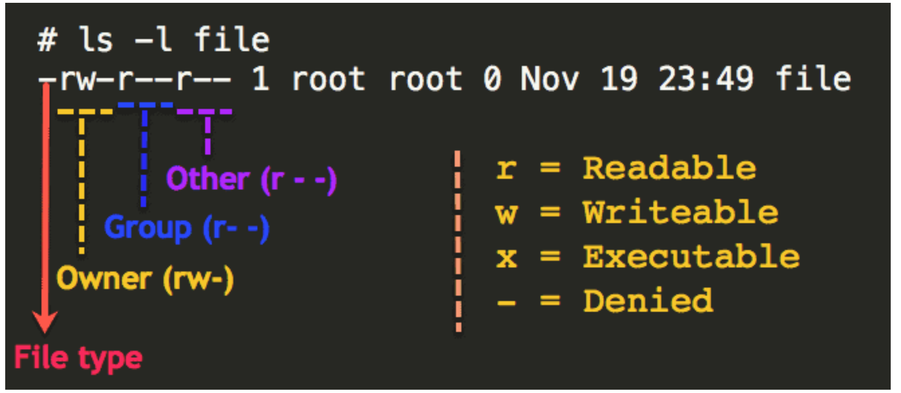
We need the ttyO1 to have read/write permission for Other, however it does not by default. Only owner (root) and group (dialout) has r/w access.
Other in this case is the Docker Node-RED container.

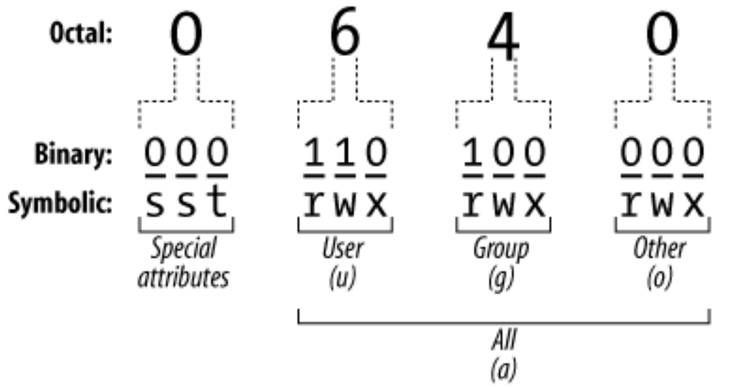
The command `chmod` allows permissions to be changed. A 6 gives rw- privileges to the device, so 666 will give owner, group and other the r/w access we need.

```
root@PFC200V3-46E709:/dev chmod 666 /dev/ttyO1
```

Another `ls /dev -l` command verifies the new permissions for Other

```
root@PFC200V3-46E709:/dev chmod 666 /dev/ttyO1
root@PFC200V3-46E709:/dev ls -l
crw-rw-rw-    1 root     dialout   246,   1 Dec 11 02:48 ttyO1
```





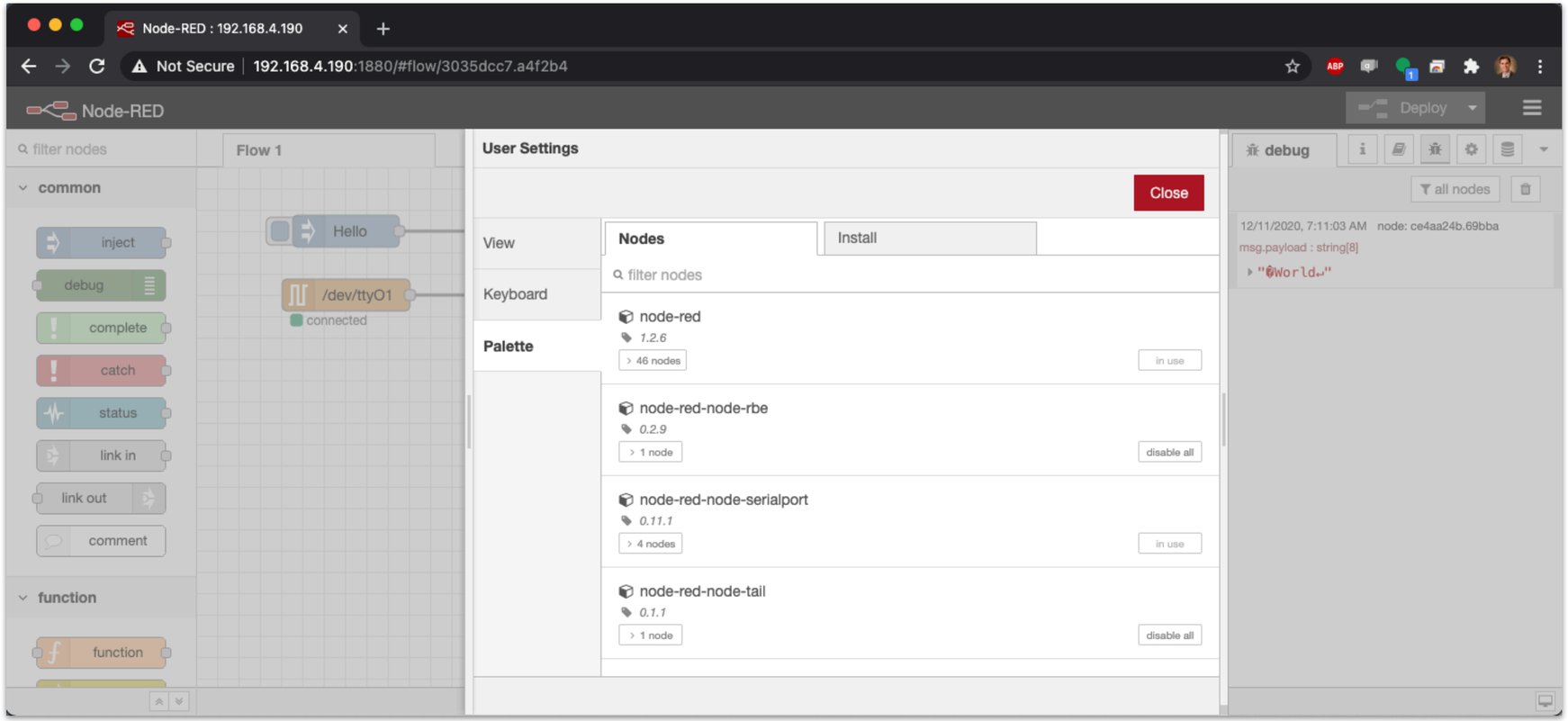
Step 3:

Install the Node-RED container with docker onto the PFC 200.

```
docker volume create node_red_user_data
docker run -d --restart unless-stopped --network=host -v node_red_user_data:/data --device=/dev/ttyO1:/dev/ttyO1:rw nodered/node-red
```

Step 4:

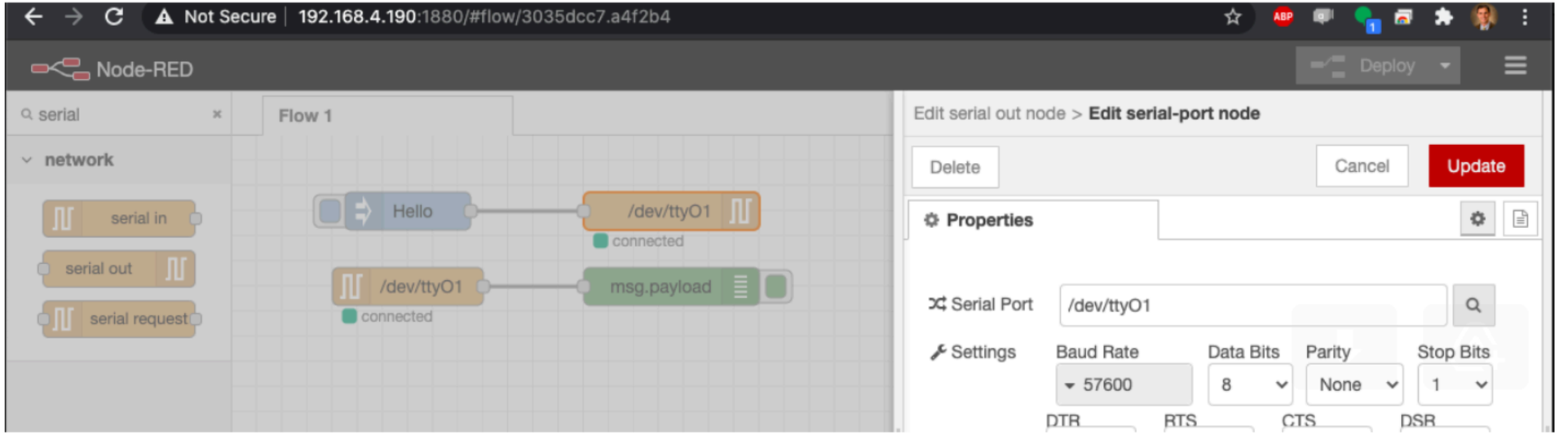
Install the Node-RED node called `node-red-serialport` in the palette manager.



Step 5:

Drag 

in the order shown below for a simple test.



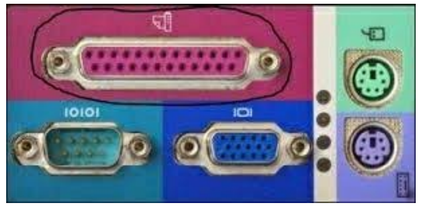
Configure the `serial-port` node for /dev/ttyO1 with 56700 Baud Rate & 8N1

Step 6:

When you deploy, the serial nodes should show connected.

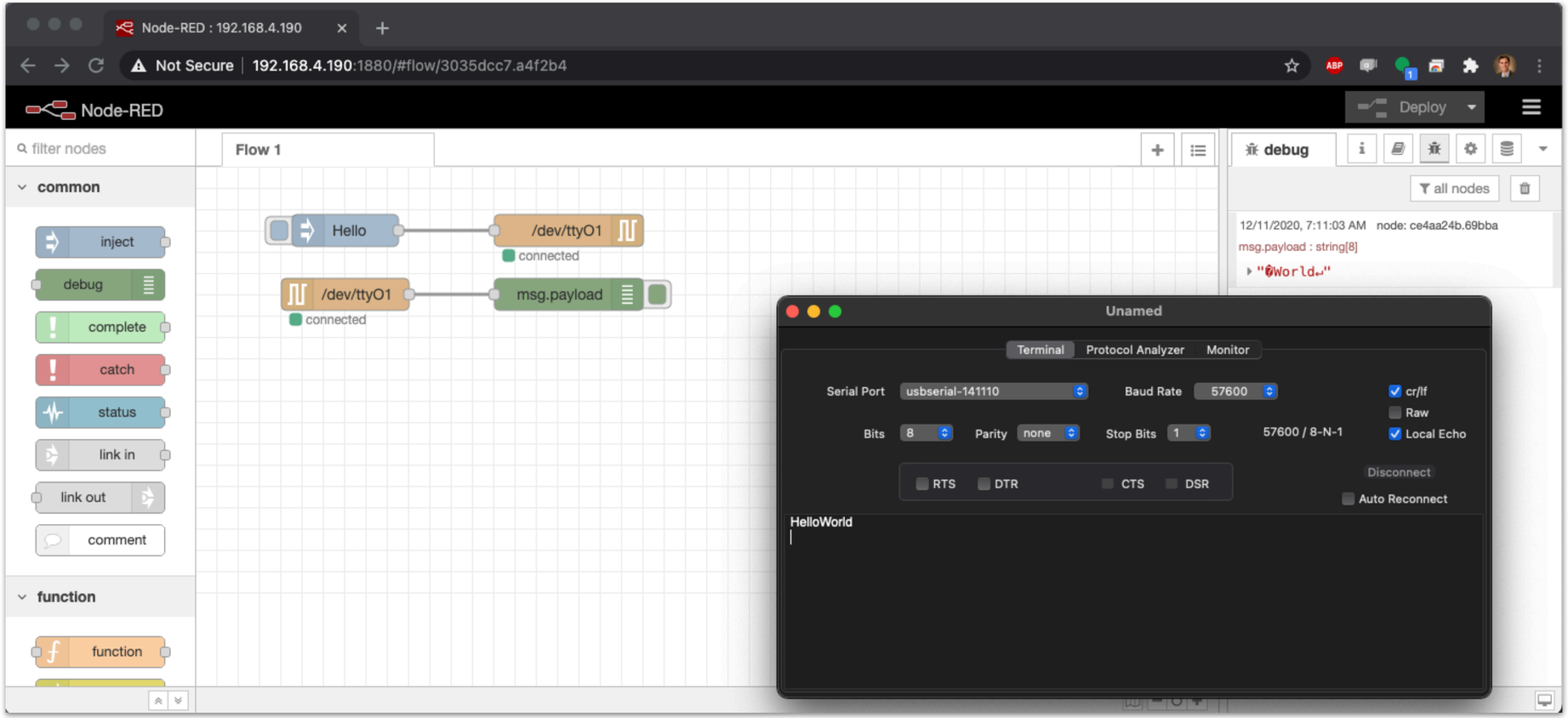
Step 7:

Connect a DB9 RS-232 cable to a computer (will a null modem inline) together with the PFC200 serial port.



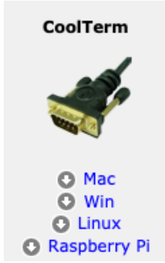
Step 8:

Using a serial terminal program on the computer, you should see ASCII in debug on both ends when you type and click the inject buttons. You now have a working serial port in Node-RED for your application development!



If you need a serial terminal program, here is an open source one called CoolTerm

https://freeware.the-meiers.org/



Troubleshooting:

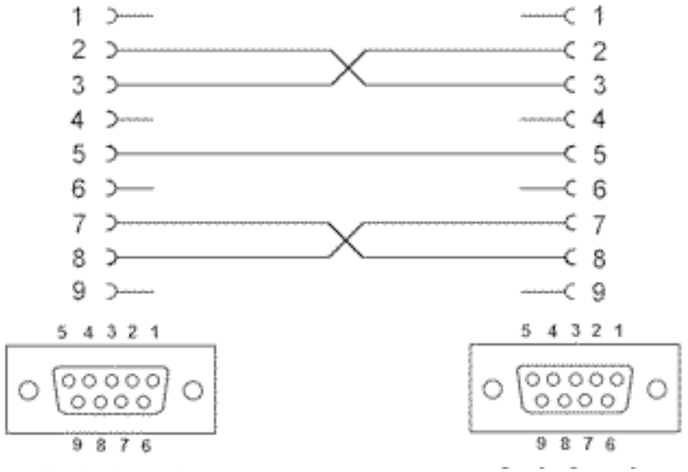
If it does not work, you can troubleshoot with the following command to start your container in interactive mode, so you can see the error messages when node-red starts.

```
root@PFC200V3-46E709:/dev docker run --rm -t --network=host -v node_red_user_data:/data --device=/dev/ttyO1:/dev/ttyO1:rw nodered/node-red

11 Dec 12:11:56 - [info] Started flows
11 Dec 12:11:56 - [error] serial port /dev/ttyO1 error: Error: Error: Permission denied, cannot open /dev/ttyO1
```

Make sure you press enter when sending strings from the computer to the PFC.

If the serial node shows connected, but ASCII characters are not showing then check the wiring.

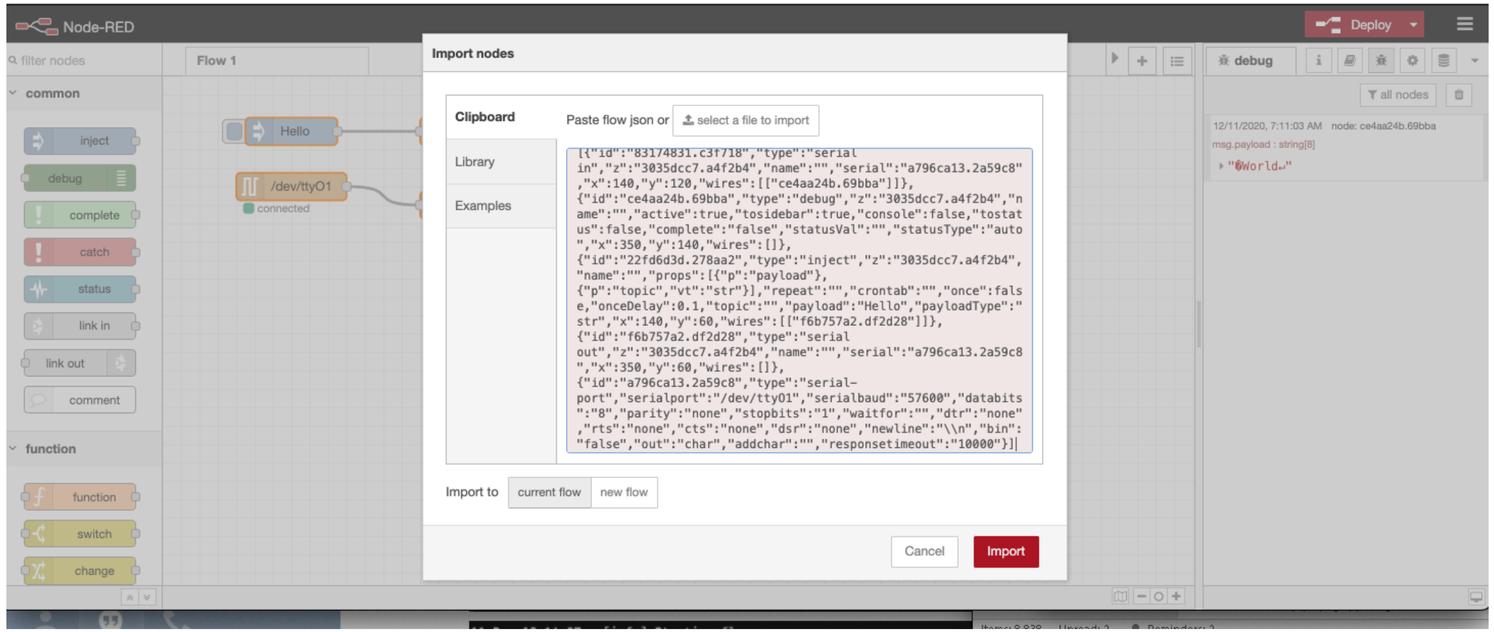


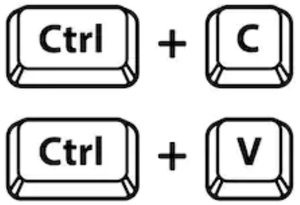
Computer Serial Port PFC 200 Serial Port

DCE to DCE Connections

DCE Device (Modem) DB9			DCE Device (Modem) DB9	
Pin# DB9 RS-232 Signal Names		Signal Direction	Pin# DB9 RS-232 Signal Names	
#1 Carrier Detector (DCD)	CD		#1 Carrier Detector (DCD)	CD
#2 Receive Data (Rx)	RD		#2 Receive Data (Rx)	RD
#3 Transmit Data (Tx)	TD		#3 Transmit Data (Tx)	TD
#4 Data Terminal Ready	DTR		#4 Data Terminal Ready	DTR
#5 Signal Ground/Common (SG)	GND		#5 Signal Ground/Common (SG)	GND
#6 Data Set Ready	DSR		#6 Data Set Ready	DSR
#7 Request to Send	RTS		#7 Request to Send	RTS
#8 Clear to Send	CTS		#8 Clear to Send	CTS
#9 Ring Indicator	RI		#9 Ring Indicator	RI
Soldered to DB9 Metal - Shield	FGND		Soldered to DB9 Metal - Shield	FGND

Here is the example flow for this test. Import with copy/paste in Node-RED.





[{"id":"83174831.c3f718","type":"serial
in","z":"3035dcc7.a4f2b4","name":"","serial":"a796ca13.2a59c8","x":140,"y":120,"wires":[["ce4aa24b.69bba"]]},
{"id":"ce4aa24b.69bba","type":"debug","z":"3035dcc7.a4f2b4","name":"","active":true,"tosidebar":true,"console":false,"tosta
tus":false,"complete":"false","statusVal":"","statusType":"auto","x":350,"y":140,"wires":[]},
{"id":"22fd6d3d.278aa2","type":"inject","z":"3035dcc7.a4f2b4","name":"","props":[{"p":"payload"},
{"p":"topic","vt":"str"}],"repeat":"","crontab":"","once":false,"onceDelay":0.1,"topic":"","payload":"Hello","payloadType":"str",
"x":140,"y":60,"wires":[["f6b757a2.df2d28"]]},{"id":"f6b757a2.df2d28","type":"serial
out","z":"3035dcc7.a4f2b4","name":"","serial":"a796ca13.2a59c8","x":350,"y":60,"wires":[]},
{"id":"a796ca13.2a59c8","type":"serial-port","serialport":"/dev/
ttyO1","serialbaud":"57600","databits":"8","parity":"none","stopbits":"1","waitfor":"","dtr":"none","rts":"none","cts":"none","d
sr":"none","newline":"\\n","bin":"false","out":"char","addchar":"","responsetimeout":"10000"}]