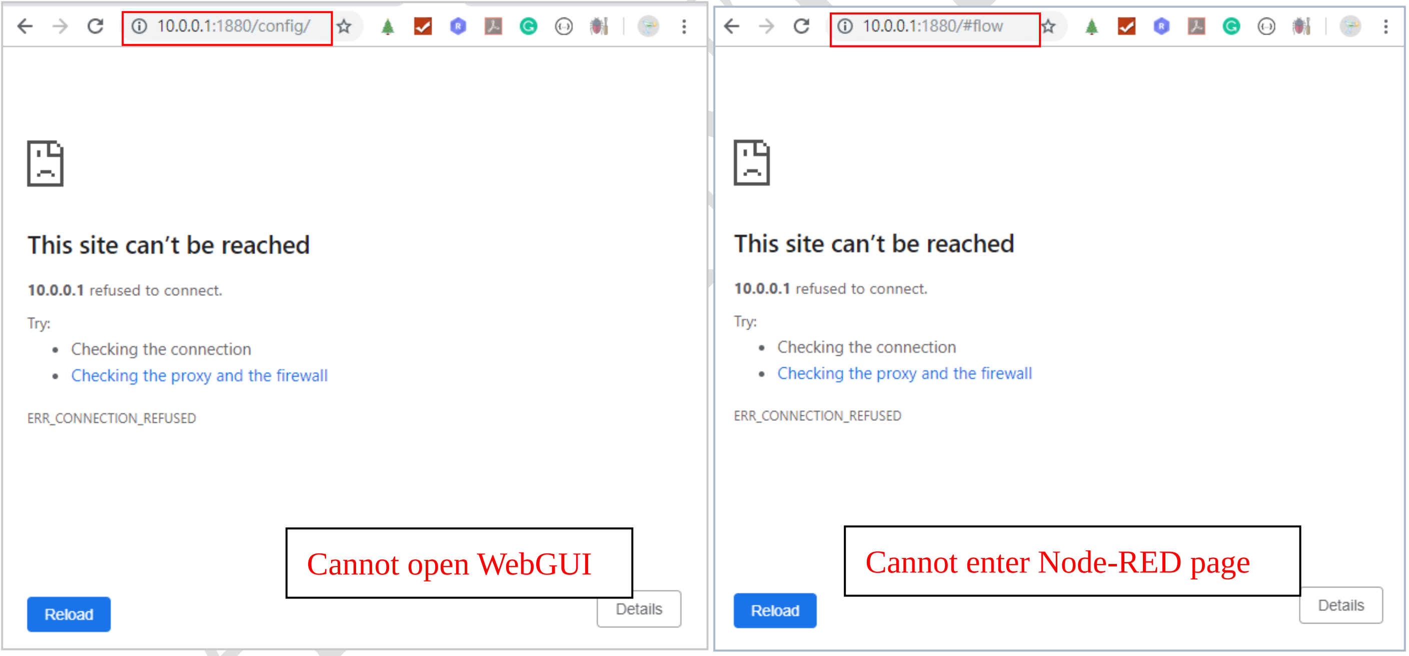
Advantech AE Technical Share Document

Date	2019/9/25	**SR#**	
Category	■FAQ □SOP	**Related OS**	N/A
Abstract	How to find the root cause for node-red server crashed		
Keyword	ADAM-6700, Node-red server, Crashed, putty, WISE-2834		
Related Product	ADAM-6717, ADAM-6750, WISE-2834		

■ Problem Description:

When using Node-red to develop IoT application on ADAM-6700, sometimes we may crash the node-red server due to programming error. When node-red server crashed, we cannot enter the Web GUI page also the node-red editor page. User will see below screenshot and cannot to any further operation on the module. This document explains the step for you to find out the possible root cause on your node-red flow



10.0.0.1:1880/config/

This site can't be reached

10.0.0.1 refused to connect.

Try:

- Checking the connection
- Checking the proxy and the firewall

ERR_CONNECTION_REFUSED

Cannot open WebGUI

10.0.0.1:1880/#flow

This site can't be reached

10.0.0.1 refused to connect.

Try:

- Checking the connection
- Checking the proxy and the firewall

ERR_CONNECTION_REFUSED

Cannot enter Node-RED page

Fig.1 Cannot Enter WebGUI page due to node-red server crash

When node-red server crashed, if you wish to find the possible root cause, please follow below instruction.

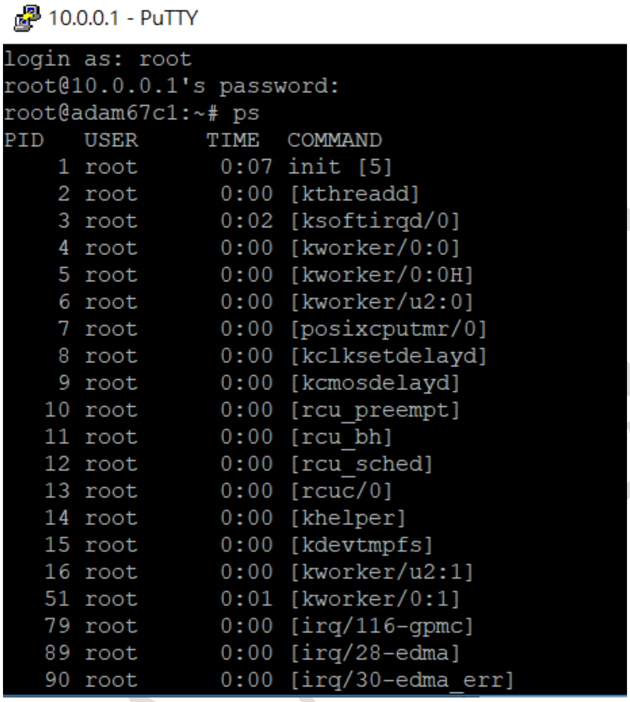
Note: You can use the same method to solve the node-red crashed problem happened on WISE-2834.

■ Solution:

Step1:

Find the node-red process in SSH

Use putty to enter ADAM-6700 module, enter “ps” command to list out all the process



```
10.0.0.1 - PuTTY
login as: root
root@10.0.0.1's password:
root@adam67c1:~# ps
PID   USER       TIME   COMMAND
    1 root       0:07 init [5]
    2 root       0:00 [kthreadd]
    3 root       0:02 [ksoftirqd/0]
    4 root       0:00 [kworker/0:0]
    5 root       0:00 [kworker/0:0H]
    6 root       0:00 [kworker/u2:0]
    7 root       0:00 [posixcputmr/0]
    8 root       0:00 [kclksetdelayd]
    9 root       0:00 [kcmosdelayd]
   10 root       0:00 [rcu_preempt]
   11 root       0:00 [rcu_bh]
   12 root       0:00 [rcu_sched]
   13 root       0:00 [rcuc/0]
   14 root       0:00 [khelper]
   15 root       0:00 [kdevtmpfs]
   16 root       0:00 [kworker/u2:1]
   51 root       0:01 [kworker/0:1]
   79 root       0:00 [irq/116-gpmc]
   89 root       0:00 [irq/28-edma]
   90 root       0:00 [irq/30-edma_err]
```

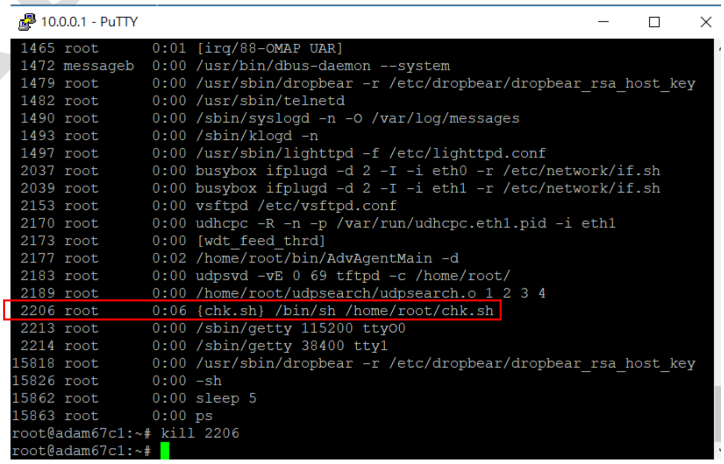
Fig.1 Enter ps command to list out all the process

Step2:

Kill the **chk.sh** process. Find the process ID via as shown in Figrue 2. User can use one of below command to kill the process.

>> Kill 2206

This is the process to make sure node-red application will keep running even the node-red process is being closed manually. The process will monitor whether node-red process is running or not, if not, it will execute the node-red process.



```
10.0.0.1 - PuTTY
 1465 root       0:01 [irq/88-OMAP UAR]
 1472 messageb   0:00 /usr/bin/dbus-daemon --system
 1479 root       0:00 /usr/sbin/dropbear -r /etc/dropbear/dropbear_rsa_host_key
 1482 root       0:00 /usr/sbin/telnetd
 1490 root       0:00 /sbin/syslogd -n -O /var/log/messages
 1493 root       0:00 /sbin/klogd -n
 1497 root       0:00 /usr/sbin/lighttpd -f /etc/lighttpd.conf
 2037 root       0:00 busybox ifplugd -d 2 -I -i eth0 -r /etc/network/if.sh
 2039 root       0:00 busybox ifplugd -d 2 -I -i eth1 -r /etc/network/if.sh
 2153 root       0:00 vsftpd /etc/vsftpd.conf
 2170 root       0:00 udhcpc -R -n -p /var/run/udhcpc.eth1.pid -i eth1
 2173 root       0:00 [wdt_feed_thrd]
 2177 root       0:02 /home/root/bin/AdvAgentMain -d
 2183 root       0:00 udpsvd -vE 0 69 tftpd -c /home/root/
 2189 root       0:00 /home/root/udpsearch/udpsearch.o 1 2 3 4
 2206 root       0:06 {chk.sh} /bin/sh /home/root/chk.sh
 2213 root       0:00 /sbin/getty 115200 ttyO0
 2214 root       0:00 /sbin/getty 38400 tty1
15818 root       0:00 /usr/sbin/dropbear -r /etc/dropbear/dropbear_rsa_host_key
15826 root       0:00 -sh
15862 root       0:00 sleep 5
15863 root       0:00 ps
root@adam67c1:~# kill 2206
root@adam67c1:~#
```

Fig.2 Kill chk.sh process

Step3:

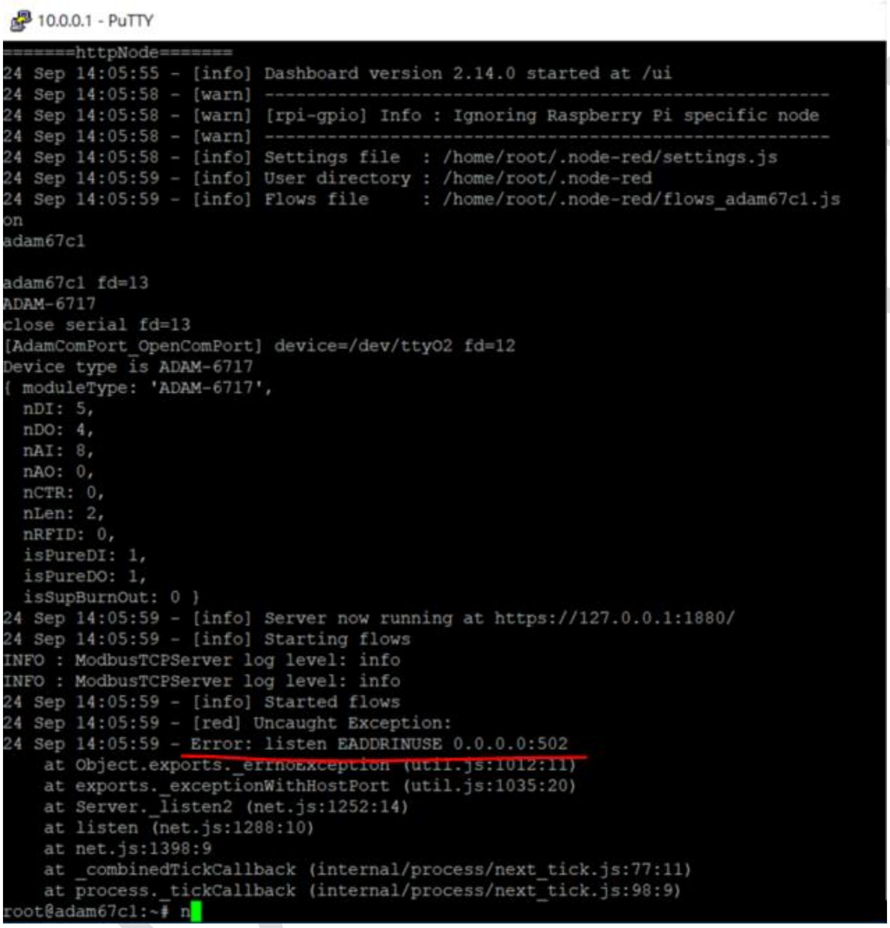
Kill node-red process, you can achieve this action by below command.

>> sh k.sh

Step4:

Run node-red program manually by below command to see the error message show on SSH screen

>> node-red &



10.0.0.1 - PuTTY

```
=======httpNode=======
24 Sep 14:05:55 - [info] Dashboard version 2.14.0 started at /ui
24 Sep 14:05:58 - [warn] ------------------------------------------------------
24 Sep 14:05:58 - [warn] [rpi-gpio] Info : Ignoring Raspberry Pi specific node
24 Sep 14:05:58 - [warn] ------------------------------------------------------
24 Sep 14:05:58 - [info] Settings file  : /home/root/.node-red/settings.js
24 Sep 14:05:59 - [info] User directory : /home/root/.node-red
24 Sep 14:05:59 - [info] Flows file     : /home/root/.node-red/flows_adam67c1.js
on
adam67c1

adam67c1 fd=13
ADAM-6717
close serial fd=13
[AdamComPort_OpenComPort] device=/dev/ttyO2 fd=12
Device type is ADAM-6717
{ moduleType: 'ADAM-6717',
  nDI: 5,
  nDO: 4,
  nAI: 8,
  nAO: 0,
  nCTR: 0,
  nLen: 2,
  nRFID: 0,
  isPureDI: 1,
  isPureDO: 1,
  isSupBurnOut: 0 }
24 Sep 14:05:59 - [info] Server now running at https://127.0.0.1:1880/
24 Sep 14:05:59 - [info] Starting flows
INFO : ModbusTCPServer log level: info
INFO : ModbusTCPServer log level: info
24 Sep 14:05:59 - [info] Started flows
24 Sep 14:05:59 - [red] Uncaught Exception:
24 Sep 14:05:59 - Error: listen EADDRINUSE 0.0.0.0:502
    at Object.exports._errnoException (util.js:1012:11)
    at exports._exceptionWithHostPort (util.js:1035:20)
    at Server._listen2 (net.js:1252:14)
    at listen (net.js:1288:10)
    at net.js:1398:9
    at _combinedTickCallback (internal/process/next_tick.js:77:11)
    at process._tickCallback (internal/process/next_tick.js:98:9)
root@adam67c1:~# n
```

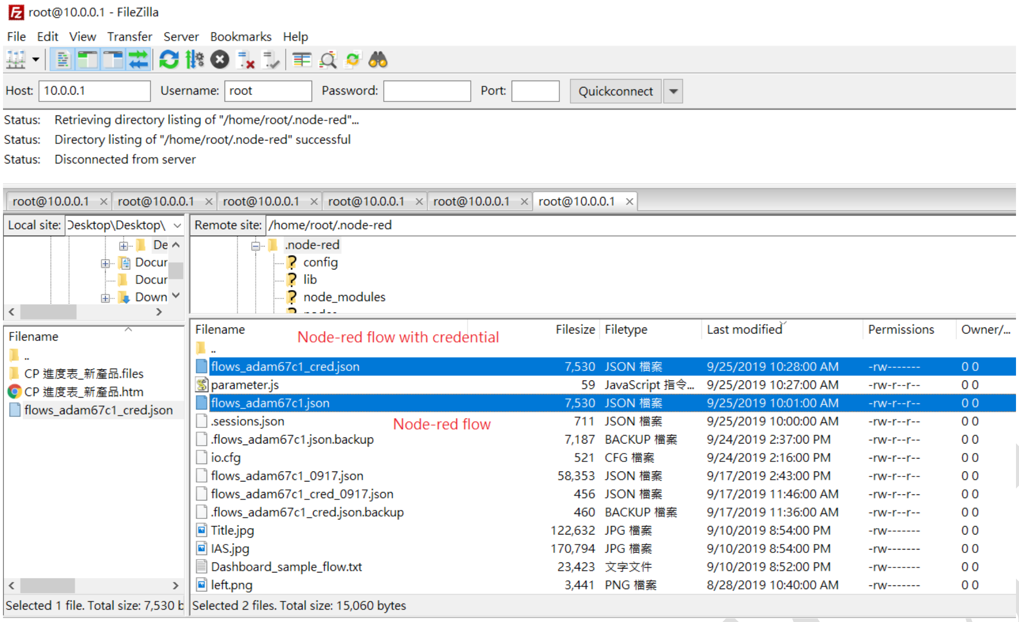
Fig.3 Error message in SSH console

In this example, the node-red flow use two modbus server node and are both running at port 502, which cause the node-red web server crashed.

Step 5:

Enter the ftp server of ADAM-6700, rename the flows_adam67c1.json and flows_adam67c1_cred.json for fixing the problem in node-red flow later.

The default account/password of ftp server is root and no password.



root@10.0.0.1 - FileZilla

Host: 10.0.0.1 Username: root Password: Port: Quickconnect

```
Status:   Retrieving directory listing of "/home/root/.node-red"...
Status:   Directory listing of "/home/root/.node-red" successful
Status:   Disconnected from server
```

Local site: Desktop\Desktop\

Remote site: /home/root/.node-red

Filename
..
CP 進度表_新產品.files
CP 進度表_新產品.htm
flows_adam67c1_cred.json

Node-red flow with credential

Node-red flow

Filename	Filesize	Filetype	Last modified	Permissions	Owner/...
..					
flows_adam67c1_cred.json	7,530	JSON 檔案	9/25/2019 10:28:00 AM	-rw-------	0 0
parameter.js	59	JavaScript 指令...	9/25/2019 10:27:00 AM	-rw-r--r--	0 0
flows_adam67c1.json	7,530	JSON 檔案	9/25/2019 10:01:00 AM	-rw-r--r--	0 0
.sessions.json	711	JSON 檔案	9/25/2019 10:00:00 AM	-rw-r--r--	0 0
.flows_adam67c1.json.backup	7,187	BACKUP 檔案	9/24/2019 2:37:00 PM	-rw-r--r--	0 0
io.cfg	521	CFG 檔案	9/24/2019 2:16:00 PM	-rw-r--r--	0 0
flows_adam67c1_0917.json	58,353	JSON 檔案	9/17/2019 2:43:00 PM	-rw-r--r--	0 0
flows_adam67c1_cred_0917.json	456	JSON 檔案	9/17/2019 11:46:00 AM	-rw-r--r--	0 0
.flows_adam67c1_cred.json.backup	460	BACKUP 檔案	9/17/2019 11:36:00 AM	-rw-r--r--	0 0
Title.jpg	122,632	JPG 檔案	9/10/2019 8:54:00 PM	-rw-------	0 0
IAS.jpg	170,794	JPG 檔案	9/10/2019 8:54:00 PM	-rw-------	0 0
Dashboard_sample_flow.txt	23,423	文字文件	9/10/2019 8:52:00 PM	-rw-------	0 0
left.png	3,441	PNG 檔案	8/28/2019 10:40:00 AM	-rw-------	0 0

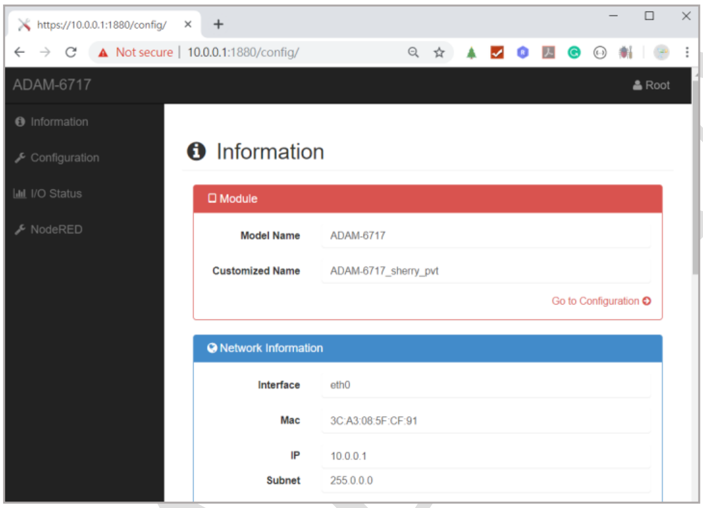
Selected 1 file. Total size: 7,530 b

Selected 2 files. Total size: 15,060 bytes

Fig.4 Rename the node-red flow in ADAM-6700

Step6:

Restart the module by a power cycle, after the system is ready, you can enter both Web GUI page and node-red editor page again as below screenshot.



https://10.0.0.1:1880/config/

Not secure | 10.0.0.1:1880/config/

ADAM-6717

Information

Module

Model Name: ADAM-6717

Customized Name: ADAM-6717_sherry_pvt

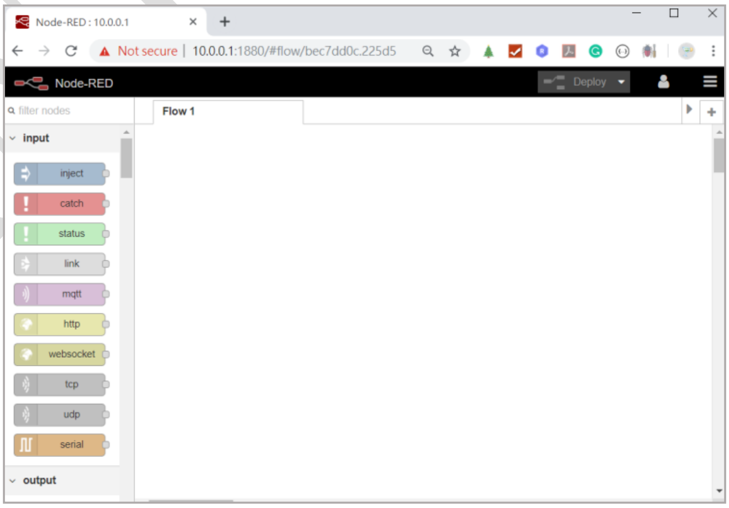
Network Information

Interface: eth0

Mac: 3C:A3:08:5F:CF:91

IP: 10.0.0.1

Subnet: 255.0.0.0



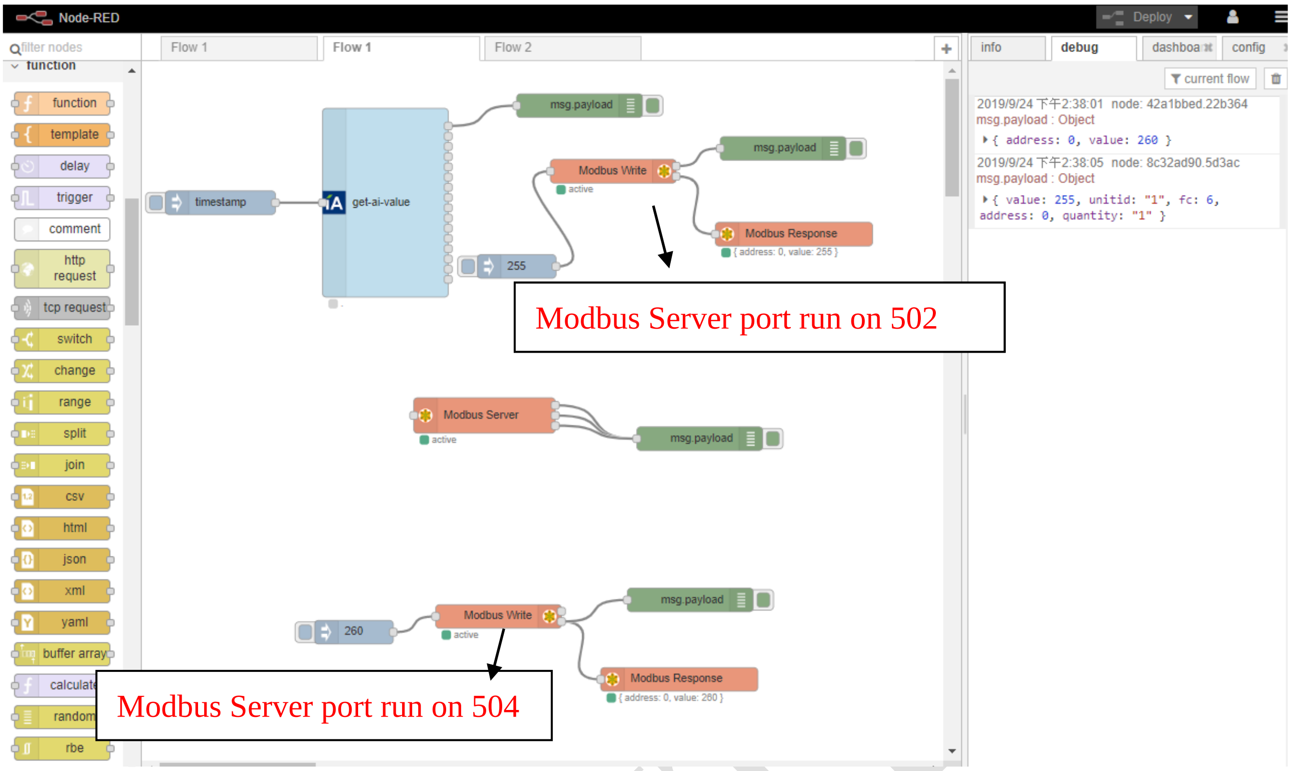
Node-RED : 10.0.0.1

Not secure | 10.0.0.1:1880/#flow/bec7dd0c.225d5

Fig.4 Both Web GUI and Node-red editor page already recovered

Step7:

Import the flows_adam67c1.json you just rename into node-red editor page, find the Modbus server node and change to different port number then click deploy to solve the programming error in node-red flow. Since the error already fixed, you can see the message show on debug column as below picture in Figure 5.



```
2019/9/24 下午2:38:01  node: 42a1bbed.22b364
msg.payload : Object
▸{ address: 0, value: 260 }
2019/9/24 下午2:38:05  node: 8c32ad90.5d3ac
msg.payload : Object
▸{ value: 255, unitid: "1", fc: 6,
address: 0, quantity: "1" }
```

Modbus Server port run on 502

Modbus Server port run on 504

Fig.5 Fixed the duplicated Modbus server port number issue.

The flow can successfully running in node-red again.