

LUA by se dala napsat i lépe, je to rozepsané pro jednotlivé topics a outputs. xxxx je ID v cloudu.

The screenshot shows the NETIO 4ALL myNetio web interface. The left sidebar contains navigation links: Outputs, M2M API Protocols (highlighted), Users, Schedules, Actions (Lua), Settings, and Log. The main content area is divided into two columns. The left column lists protocols: SNMP, Telnet/KSHELL, MQTT (highlighted), JSON API, XML API, URL API, and Modbus/TCP. The right column displays the MQTT configuration settings:

- ☒ Enable MQTT
- MQTT status: Connected (Connected)
- MQTT mode: Generic
- Broker Host: xxxx.messaging.internetc ✓
- Broker Port: 1883 ✓
- ☒ Use credentials
  - Username: use-token-auth
  - Password: ..... [Show password](#)
- ☐ Use SSL
  - ☒ Validate server's SSL certificate
- Client Id: d:xxxx:netio4all:01 ✓
- ☒ MQTT used only with LUA Actions
- Update period: 0 seconds
- [Save Changes](#)

Broker host: xxxx.messaging.internetofthings.ibmcloud.com

mqtt Publish

mqtt Subscribes

mqtt set-state outs 1-4

Create Rule

☒ Enable rule

Name: mqtt Publish ✓

Description: connection to bluemix

Trigger: System variables updated ▼

Schedule: Always ▼

```

-- voltage
local topic = "iot-2/evt/voltage/fmt/format_string"
local report = devices.system.voltage

devices.system.MqttPublish{topic=topic, payload=string.format(report)}

-- frequency
local topic = "iot-2/evt/frequency/fmt/format_string"
local report = devices.system.frequency

devices.system.MqttPublish{topic=topic, payload=string.format(report)}

-- OUTPUT1
local output = 1

local topic = "iot-2/evt/out"..output.."current/fmt/format_string"
local report = devices.system["output" .. output .. "_current"]
devices.system.MqttPublish{topic=topic, payload=string.format(report)}

local topic = "iot-2/evt/out"..output.."powerFactor/fmt/format_string"
local report = devices.system["output" .. output .. "_powerFactor"]
devices.system.MqttPublish{topic=topic, payload=string.format(report)}

-- OUTPUT2
output = 2

```

```

-- voltage
    local topic = "iot-2/evt/voltage/fmt/format_string"
    local report = devices.system.voltage

    devices.system.MqttPublish{topic=topic, payload=string.format(report)}

-- frequency
    local topic = "iot-2/evt/frequency/fmt/format_string"
    local report = devices.system.frequency

    devices.system.MqttPublish{topic=topic, payload=string.format(report)}

-- OUTPUT1
local output = 1

    local topic = "iot-2/evt/out"..output.."current/fmt/format_string"
    local report = devices.system["output" .. output .. "_current"]
    devices.system.MqttPublish{topic=topic, payload=string.format(report)}

    local topic = "iot-2/evt/out"..output.."powerFactor/fmt/format_string"
    local report = devices.system["output" .. output .. "_powerFactor"]
    devices.system.MqttPublish{topic=topic, payload=string.format(report)}

-- OUTPUT2
output = 2

```

```
local topic = "iot-2/evt/out"..output.."current/fmt/format_string"
local report = devices.system["output" .. output .. "_current"]
devices.system.MqttPublish{topic=topic, payload=string.format(report)}
```

```
local topic = "iot-2/evt/out"..output.."powerFactor/fmt/format_string"
local report = devices.system["output" .. output .. "_powerFactor"]
devices.system.MqttPublish{topic=topic, payload=string.format(report)}
```

-- OUTPUT3

output = 3

```
local topic = "iot-2/evt/out"..output.."current/fmt/format_string"
local report = devices.system["output" .. output .. "_current"]
devices.system.MqttPublish{topic=topic, payload=string.format(report)}
```

```
local topic = "iot-2/evt/out"..output.."powerFactor/fmt/format_string"
local report = devices.system["output" .. output .. "_powerFactor"]
devices.system.MqttPublish{topic=topic, payload=string.format(report)}
```

-- OUTPUT4

output = 4

```
local topic = "iot-2/evt/out"..output.."current/fmt/format_string"
local report = devices.system["output" .. output .. "_current"]
devices.system.MqttPublish{topic=topic, payload=string.format(report)}
```

```
local topic = "iot-2/evt/out"..output.."powerFactor/fmt/format_string"
local report = devices.system["output" .. output .. "_powerFactor"]
devices.system.MqttPublish{topic=topic, payload=string.format(report)}
```

mqtt Publish

mqtt Subscribes

mqtt set-state outs 1-4

Create Rule

☒ Enable rule

Name: mqtt Subscribes ✓

Description:

Trigger: Mqtt has connected to the b ▼

Schedule: Always ▼

```

-- OUTPUT 1
local topic1 = "iot-2/cmd/out1action/fmt/format_string"
devices.system.MqttLuaSubscribe{topic=topic1}
logf("Subscribed to: %s", topic1)

-- OUTPUT 2
local topic2 = "iot-2/cmd/out2action/fmt/format_string"
devices.system.MqttLuaSubscribe{topic=topic2}
logf("Subscribed to: %s", topic2)

-- OUTPUT 3
local topic3 = "iot-2/cmd/out3action/fmt/format_string"
devices.system.MqttLuaSubscribe{topic=topic3}
logf("Subscribed to: %s", topic3)

-- OUTPUT 4
local topic4 = "iot-2/cmd/out4action/fmt/format_string"
devices.system.MqttLuaSubscribe{topic=topic4}
logf("Subscribed to: %s", topic4)

```

```

-- OUTPUT 1
local topic1 = "iot-2/cmd/out1action/fmt/format_string"
devices.system.MqttLuaSubscribe{topic=topic1}
logf("Subscribed to: %s", topic1)

-- OUTPUT 2
local topic2 = "iot-2/cmd/out2action/fmt/format_string"
devices.system.MqttLuaSubscribe{topic=topic2}
logf("Subscribed to: %s", topic2)

-- OUTPUT 3
local topic3 = "iot-2/cmd/out3action/fmt/format_string"
devices.system.MqttLuaSubscribe{topic=topic3}
logf("Subscribed to: %s", topic3)

-- OUTPUT 4
local topic4 = "iot-2/cmd/out4action/fmt/format_string"
devices.system.MqttLuaSubscribe{topic=topic4}
logf("Subscribed to: %s", topic4)

```

mqtt Publish ✖

mqtt Subscribes ✖

**mqtt set-state  
outs 1-4** ✖

Create Rule

☒ Enable rule

Name: mqtt set-state outs 1-4 ✓

Description:

Trigger: Mqtt received message on si ▼

Schedule: Always ▼

```

local topic = event.args.topic
local payload = event.args.payload
local report

logf("mqtt received payload '%s' on topic '%s'", payload, topic)

-- OUTPUT 1
if topic == "iot-2/cmd/out1action/fmt/format_string" then
devices.system.SetOut{output=1,value=payload}
  -- report state
  topic = "iot-2/evt/out1state/fmt/format_string"
  report = devices.system["output1_state"]
  devices.system.MqttPublish{topic=topic, payload=string.format(report)}
end

-- OUTPUT 2
if topic == "iot-2/cmd/out2action/fmt/format_string" then
devices.system.SetOut{output=2,value=payload}
  topic = "iot-2/evt/out2state/fmt/format_string"
  report = devices.system["output2_state"]
  devices.system.MqttPublish{topic=topic, payload=string.format(report)}
end

-- OUTPUT 3
if topic == "iot-2/cmd/out3action/fmt/format_string" then

```

```

local topic = event.args.topic
local payload = event.args.payload
local report

```

```

logf("mqtt received payload '%s' on topic '%s'", payload, topic)

```

```

-- OUTPUT 1
if topic == "iot-2/cmd/out1action/fmt/format_string" then
devices.system.SetOut{output=1,value=payload}
  -- report state
  topic = "iot-2/evt/out1state/fmt/format_string"
  report = devices.system["output1_state"]
  devices.system.MqttPublish{topic=topic, payload=string.format(report)}
end

```

```

-- OUTPUT 2
if topic == "iot-2/cmd/out2action/fmt/format_string" then
devices.system.SetOut{output=2,value=payload}
  topic = "iot-2/evt/out2state/fmt/format_string"
  report = devices.system["output2_state"]
  devices.system.MqttPublish{topic=topic, payload=string.format(report)}
end

```

```

-- OUTPUT 3
if topic == "iot-2/cmd/out3action/fmt/format_string" then

```

```

devices.system.SetOut{output=3,value=payload}
    topic = "iot-2/evt/out3state/fmt/format_string"
    report = devices.system["output3_state"]
    devices.system.MqttPublish{topic=topic, payload=string.format(report)}
end

```

-- OUTPUT 4

```

if topic == "iot-2/cmd/out4action/fmt/format_string" then
devices.system.SetOut{output=4,value=payload}
    topic = "iot-2/evt/out4state/fmt/format_string"
    report = devices.system["output4_state"]
    devices.system.MqttPublish{topic=topic, payload=string.format(report)}
end

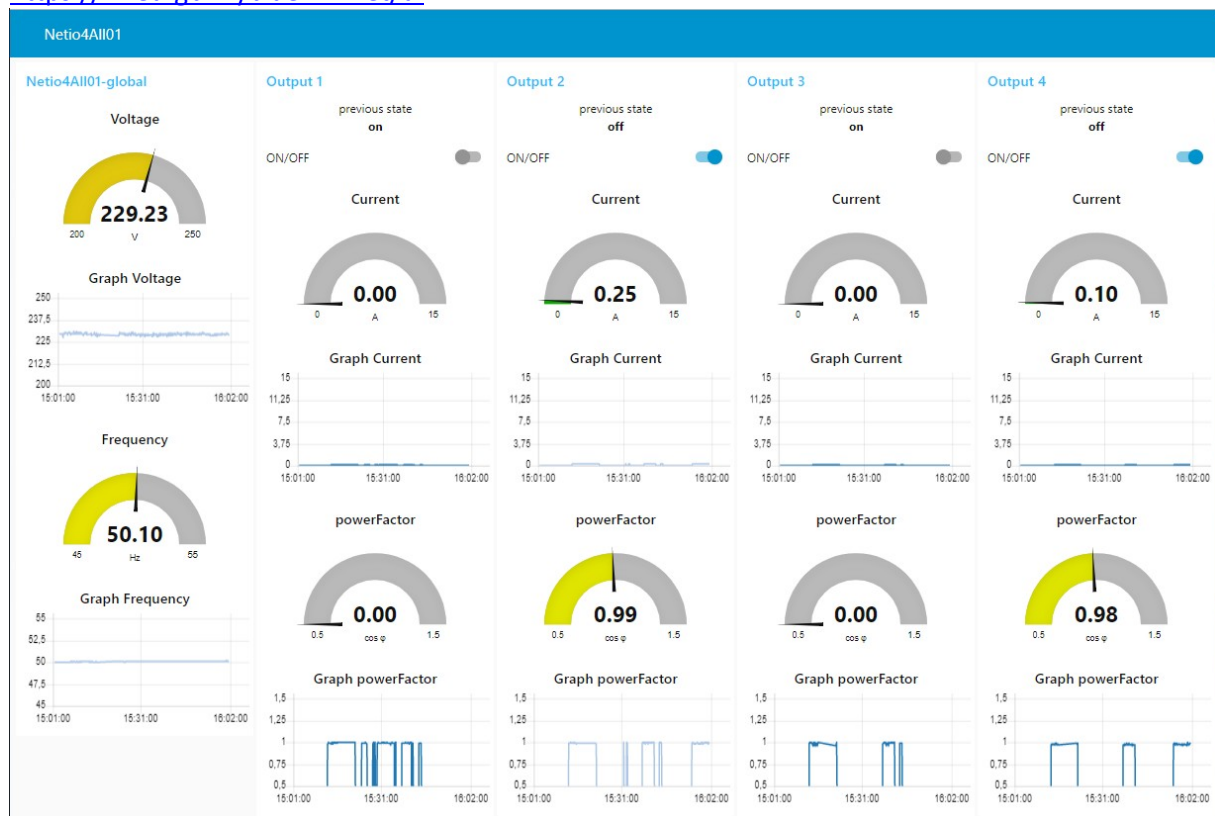
```

(publikování stavu outputX\_state ale odpovídá předchozímu stavu před provedením scriptu)

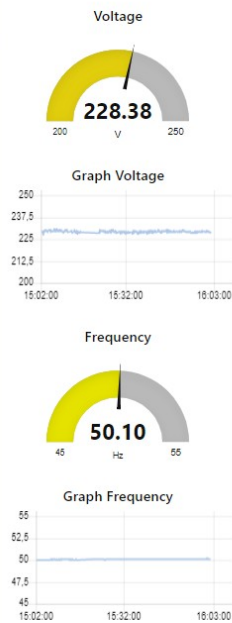
### konfigurační soubory LUA:

mqtt\_publish.txt  
mqtt\_subscribe.txt  
mqtt\_set-state.txt

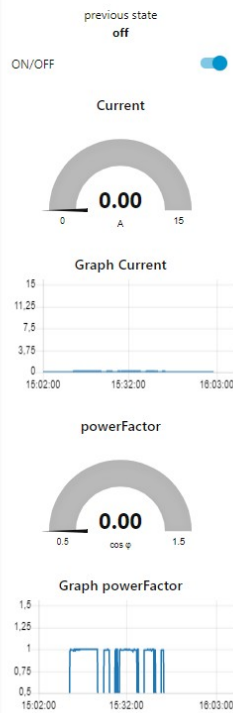
<https://xx.eu-gb.mybluemix.net/ui>



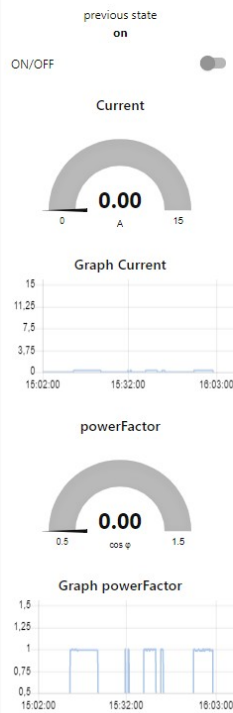
## Netio4All01-global



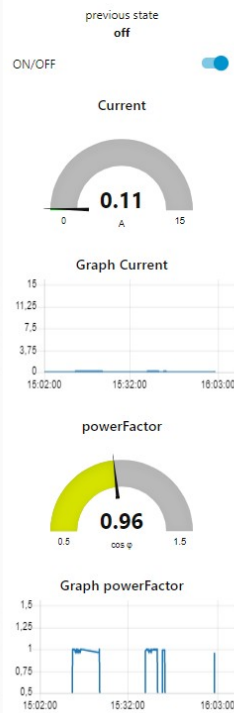
## Output 1



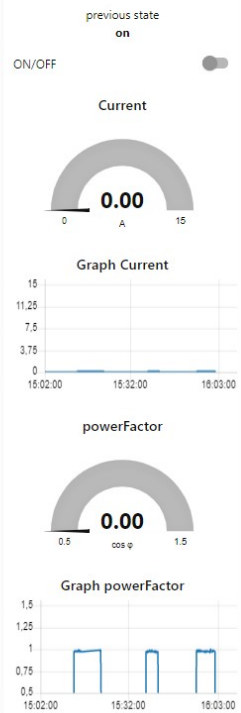
## Output 2

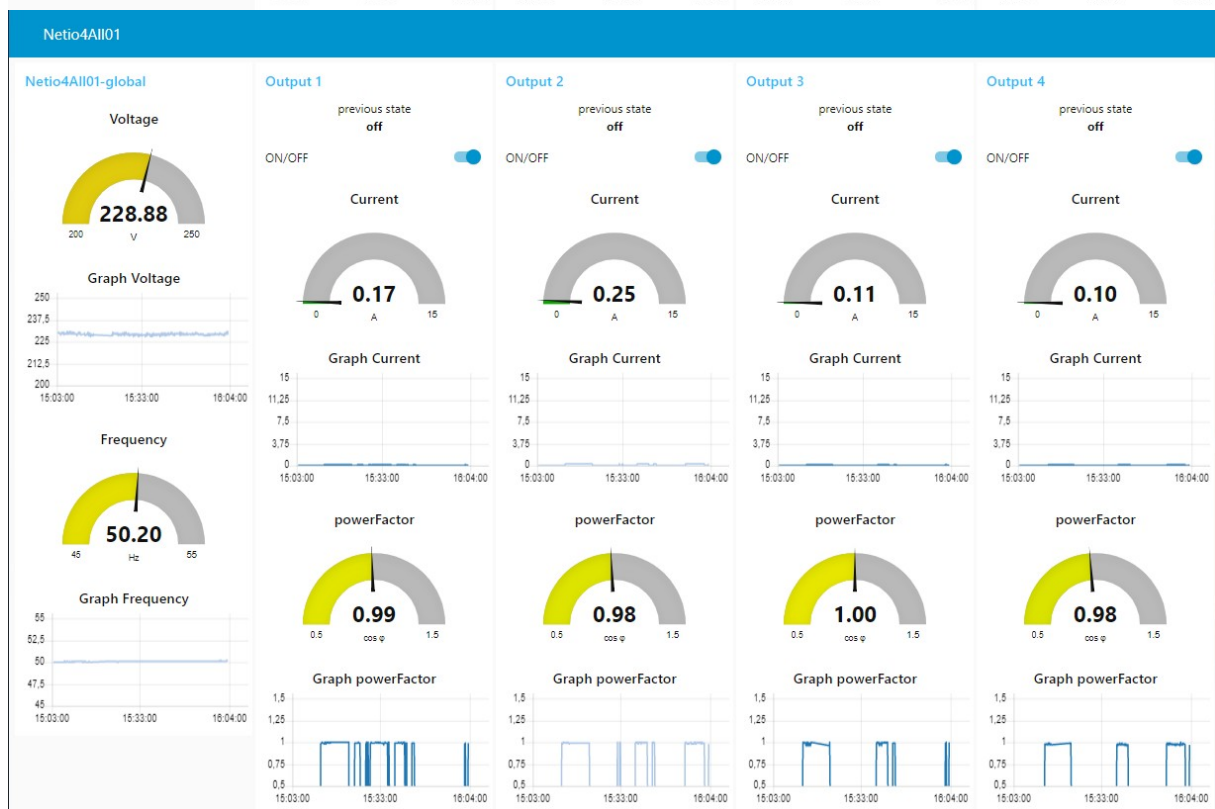
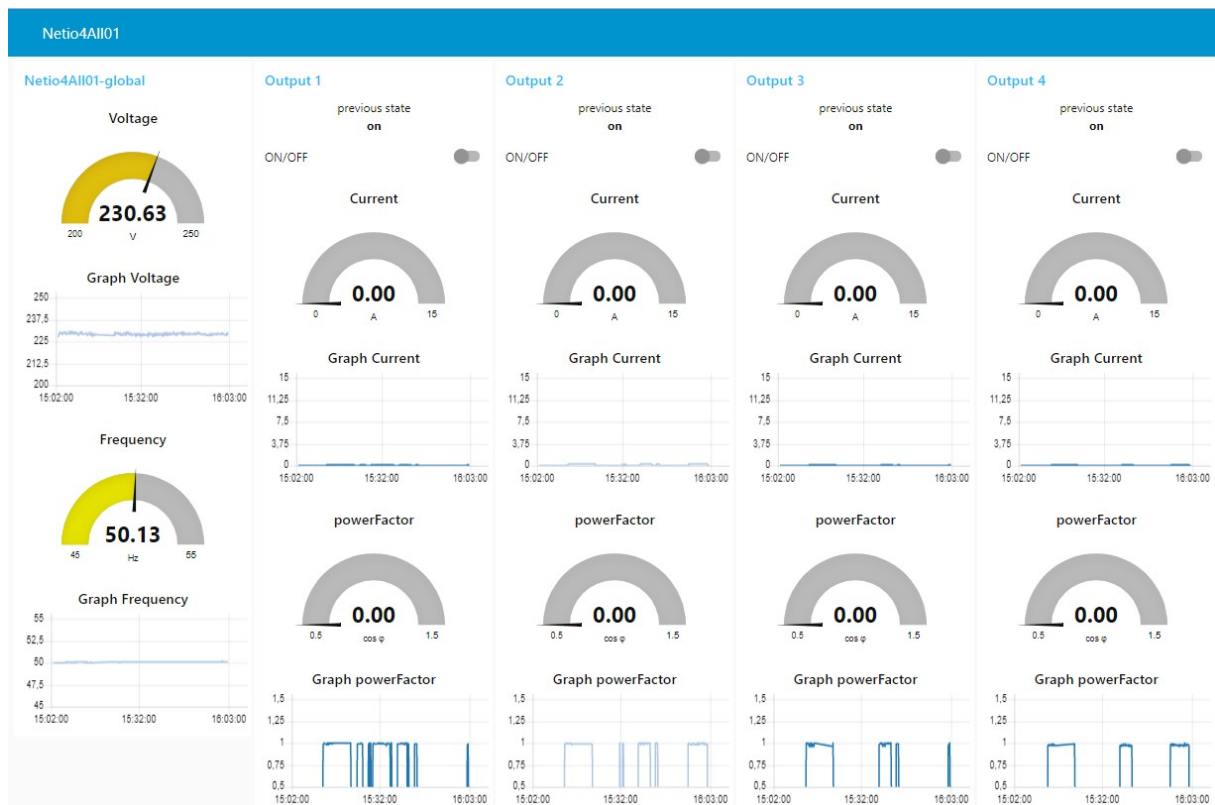


## Output 3



## Output 4





konfigurační soubory flow Node-RED:

Netio4All01-global.json

n4a01-Out1.json

n4a01-Out2.json

n4a01-Out3.json

n4a01-Out4.json