

TimePeace IOT!

Cognitive Assistance Technology



Internet of Things Applied to time & location
Using a Raspberry Pi

To offer reassurance in the early stages of Alzheimer's disease

Find us & join in!

Search google for "Raspberry Pint London"
or click on

<https://www.meetup.com/Raspberry-Pint-London/>

<https://www.facebook.com/groups/raspberrypint/>

Presentation to "Raspberry Pint London"
28th February 2017 by David Penney

Definition:

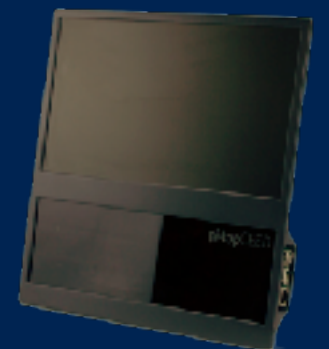
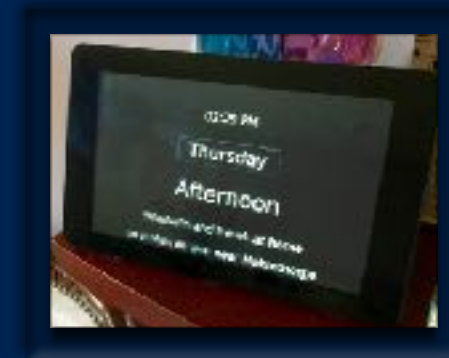
Cognitive Assistance Technology

“use of technology (usually high tech) to augment and assist cognitive processes such as attention, **memory**, self-regulation, navigation, emotion recognition and management, planning, and sequencing activity”

Source: LoPresti, E.F., Mihailidis, A. & Kirsch, N. (2004).
Assistive Technology for cognitive rehabilitation: State of the art.
Neuropsychological Rehabilitation, 14, 5-39

3 projects to reassure a relative diagnosed with early stage Alzheimer's disease

- ✧ TimePeace Clock shows time of day & day of week
- ✧ TimePeace AlexaPi ask who is visiting
- ✧ TimePeace CalendarTV see who is visiting



Raspberry Pint London: Project of the year 2017

[https://github.com/dpenney5/Raspberry-Pint-London/blob/master/TimePeace Presentation to Raspberry Pint London 29 Nov 2016.pdf](https://github.com/dpenney5/Raspberry-Pint-London/blob/master/TimePeace%20Presentation%20to%20Raspberry%20Pint%20London%2029%20Nov%202016.pdf)

Last time I spoke about “Potential next projects”

- ✦ Link the AlexaPi to the CalendarTV with NodeRed/MQTT
 - ✦ Send common commands using the CalendarTV remote control to the AlexaPi
 - ✦ Common commands from remote control
“What day is it today?”, “What is happening today?”
 - ✦ Possibly also apply Alexa Skills to use voice via AlexaPi to control CalendarTV & TimePeace using Node-Red
- ✦ An easy to manipulate timeline of personal history & photos in the context of historical events, music, TV, Film

Internet of Things Infrastructure

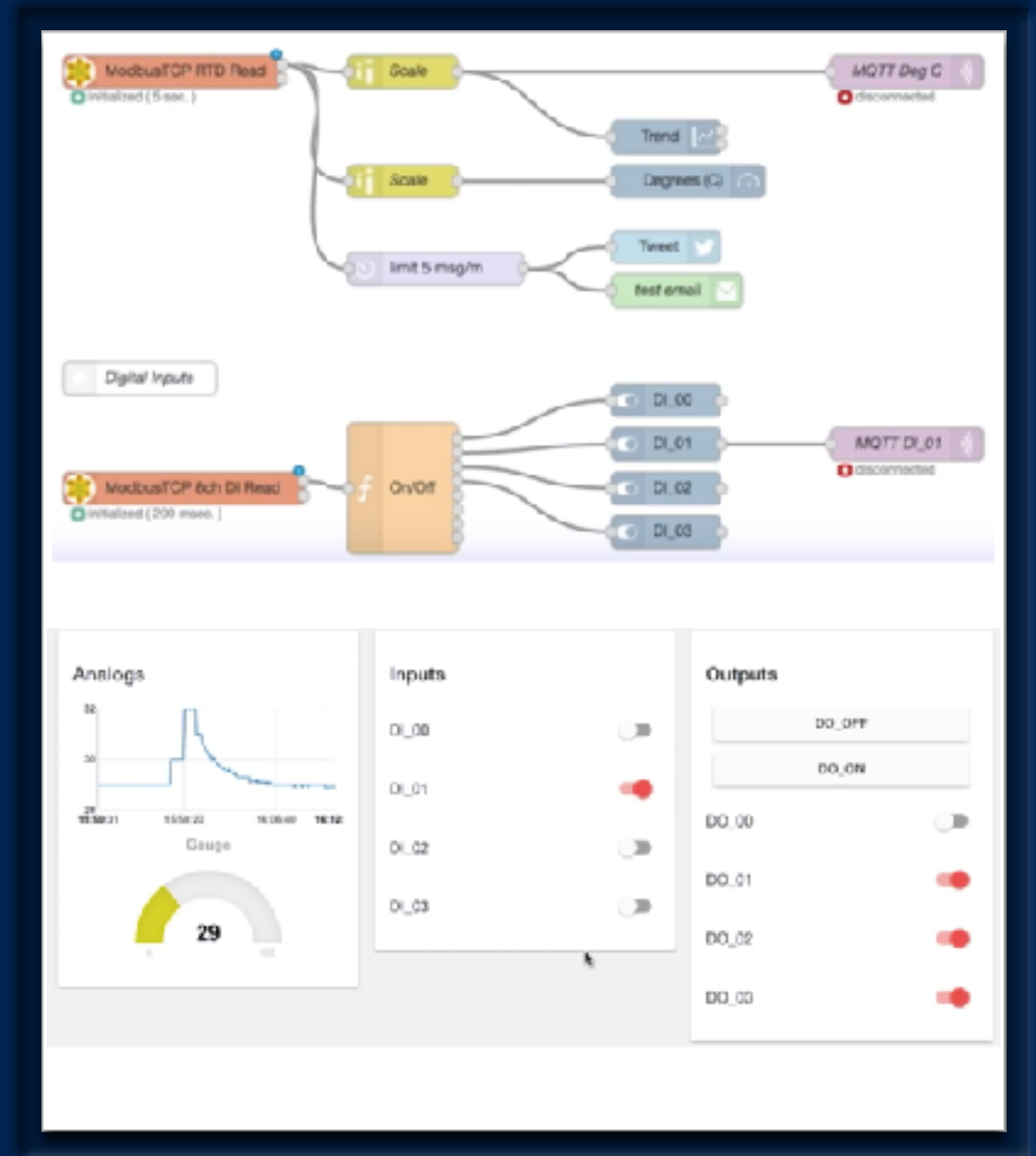
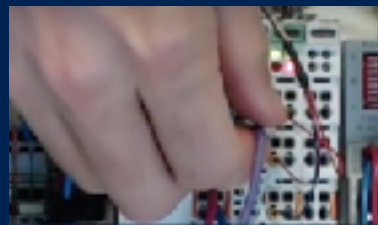
- ✧ Paho MQTT Mosquito

- ✧ Easy to use
- ✧ Publish & Subscribe
- ✧ Fast status & update sharing



- ✧ IBM Node Red

- ✧ Easy to use
- ✧ Part of Raspbian distribution
- ✧ Rapid delivery
- ✧ Graphical orchestration
- ✧ Build status web pages
- ✧ Save data
- ✧ Link systems & coordinate activities



Task at hand is to add MQTT (Mosquitto) & a Node-Red Dashboard

- ✦ One computer is designated as a broker node to act as the central clearing house for topic updates (Easy)
- ✦ Define the topic structure for publication (Critical)
- ✦ Methodically instrument applications (Painstaking)
- ✦ Design a Node-Red test schematic (Easy - its installed in every Raspberry Pi & works everywhere - or you can use the IBM Blue-Mix Cloud (a global MQTT Broker :))

Methodically Test

- ✦ Used Node-Red to test messages
- ✦ Install Node-Red package for Dashboards
- ✦ Built a Dashboard
- ✦ Once you figure it out - takes a minute to drag & drop a dashboard & one click deploys it live!

MQTT load modules

```
from memcache import Client  
import paho.mqtt.client as mqtt
```


MQTT Define Parameters

```
from memcache import Client
import paho.mqtt.client as mqtt
```

```
# Define Variables for MQTT
```

```
MQTT_BROKER = "dev10.local" # dev10.local"
```

```
MQTT_PORT = 1883
```

```
MQTT_KEEPALIVE_INTERVAL = 45
```

```
MQTT_Device="%s/" % socket.gethostname() # this should be changed to use the node name
```

```
MQTT_Application="Application/TimePeace-Clock/"
```

```
#Topics
```

```
MQTT_Status_TOPIC = "Status/" + MQTT_Device # Status/dev10
```

```
MQTT_State_TOPIC = "State/" + MQTT_Device + MQTT_Application # State/dev10/Clock
```

```
MQTT_Command_TOPIC = "Command/" + MQTT_Device + MQTT_Application
```

```
MQTT_Metrics_TOPIC = "Metrics/" + MQTT_Device + devname + "/" # Metrics/dev10/wlan0/Rx
```

MQTT Initialise

```
# Define MQTT on_connect event Handler
```

```
def on_connect(mosq, obj, rc):  
    print "Connected to MQTT Broker"  
    return
```

```
# Define MQTT on_publish event Handler
```

```
def on_publish(client, userdata, mid):  
    print "Message Published..."  
    return
```

```
# Initiate MQTT Client
```

```
mqttc = mqtt.Client()
```

```
# Register Event Handlers
```

```
mqttc.on_publish = on_publish  
mqttc.on_connect = on_connect
```

```
# Connect with MQTT Broker
```

```
mqttc.connect_async(MQTT_BROKER, MQTT_PORT, MQTT_KEEPALIVE_INTERVAL)  
mqttc.loop_start()
```

```
# Publish message to MQTT Topic
```

```
print "MQTT:: Starting up on ",MQTT_Status_TOPIC  
mqttc.publish(MQTT_Status_TOPIC,"Starting up")
```

MQTT Publish information

```
ldt = datetime.datetime.now()
mqttc.publish(MQTT_State_TOPIC,
              '%s %s %s'
              % (self.time,
                 timesofday[ldt.hour],
                 ldt.strftime('%A %d-%B-%Y')))
```

Topic:

State/dev10/Clock/

Payload:

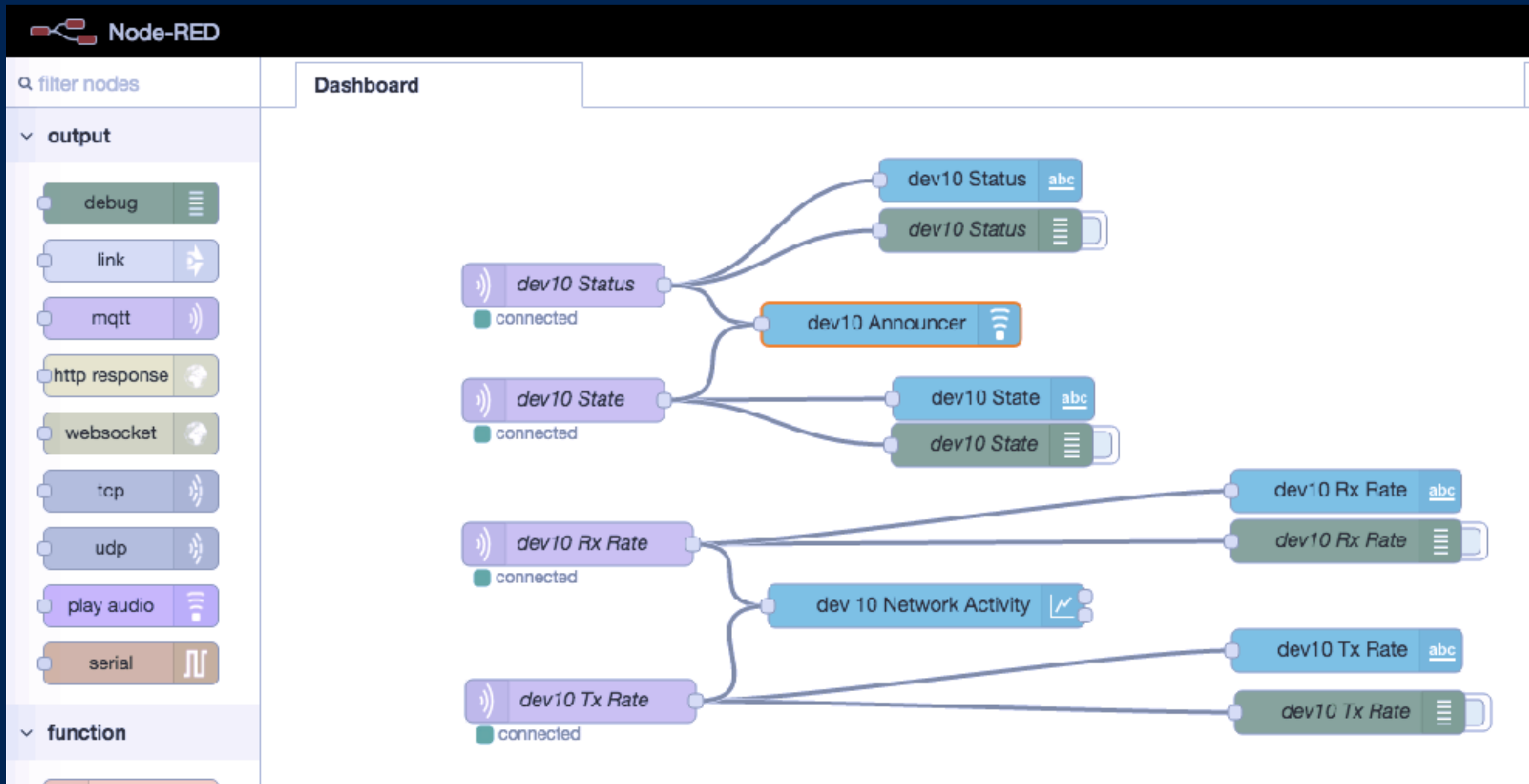
“10:38 Nearly lunchtime Tuesday 28-February-2017”

MQTT Close Down

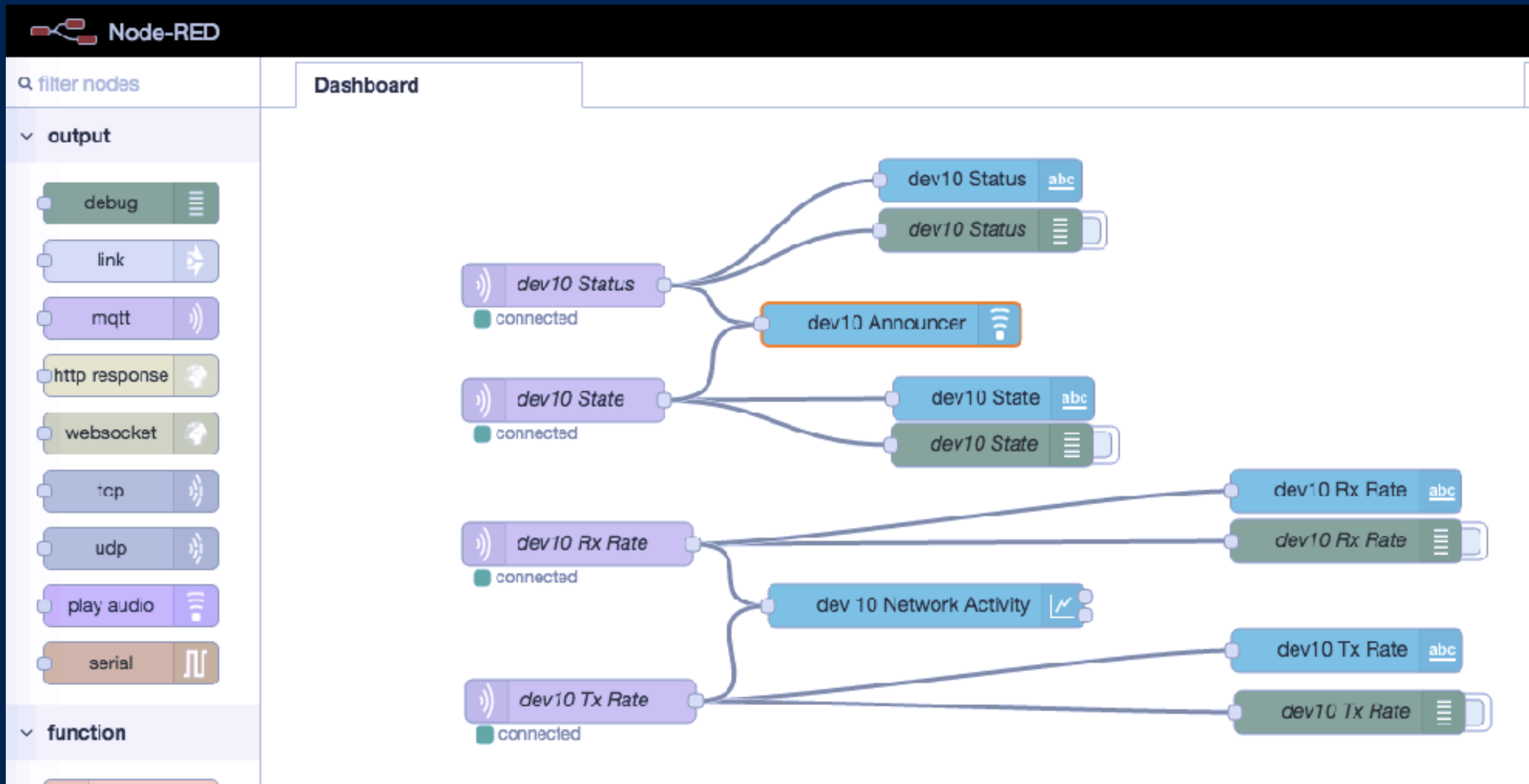
```
mqttc.publish(MQTT_Status_TOPIC,"Stopping")  
mqttc.loop_stop()
```

```
# Disconnect from MQTT_Broker  
mqttc.disconnect()
```


Node-Red Dashboard - Port 1880



Node-Red Dashboard



Node-Red Dashboard

The image displays the Node-Red web interface. A modal window titled "Edit mqtt in node" is open, showing the configuration for an MQTT node named "dev10 Status". The configuration includes:

- Server:** dev10.local:1883
- Topic:** Status/dev10/#
- QoS:** 2
- Name:** dev10 Status

The modal also features "Delete", "Cancel", and "Done" buttons. In the background, a dashboard flow is visible. It includes two MQTT nodes for "dev10 Rx Rate" and "dev10 Tx Rate", both marked as "connected". These nodes are connected to a "dev 10 Network Activity" node, which is a gauge. The flow also includes several other nodes, such as "dev10 Status", "dev10 State", and "dev10 Rx Rate", each with a corresponding gauge or display. The left sidebar shows various input and output nodes like "udp", "play audio", "serial", and "function".

Node-Red Dashboard

The screenshot displays the Node-RED web interface. On the left, a sidebar contains a search bar labeled 'filter nodes' and two categories of nodes: 'output' and 'function'. The 'output' category is expanded, showing nodes like 'debug', 'link', 'mqtt', 'http response', 'websocket', 'tcp', 'udp', 'play audio', and 'serial'. The 'function' category is also visible at the bottom.

The main workspace shows a flow with several nodes. A 'dev10 Status' node is selected, and its configuration dialog is open. The dialog has a title bar 'Edit text node' and a tab 'dev10 Status abc'. It includes 'Delete', 'Cancel', and 'Done' buttons. The configuration fields are as follows:

- Group:** A dropdown menu set to 'dev10 [Network Activity]' with an edit icon.
- Size:** A dropdown menu set to 'auto'.
- Label:** A text input field containing 'Status'.
- Value format:** A text input field containing '{{msg.payload}}'.
- Layout:** A visual representation of the node's layout, showing five 'label value' components arranged in two rows (three in the top row, two in the bottom row).
- Name:** A text input field at the bottom containing 'dev10 Status'.

In the background, other nodes in the flow are visible, including 'dev10 Rx Rate abc', 'dev10 Rx Rate abc', 'dev10 Tx Rate abc', and 'dev10 Tx Rate abc'.

Node-Red Dashboard

The image shows the Node-RED web interface. On the left, the 'output' category is expanded, showing nodes like debug, link, mqtt, http response, websocket, tcp, udp, play audio, and serial. The 'function' category is also visible at the bottom. The main workspace is titled 'Dashboard' and contains several MQTT nodes. A dialog box titled 'Edit mqtt in node' is open, showing the configuration for a node named 'dev10 Tx Rate'. The dialog has a 'Delete' button on the left and 'Cancel' and 'Done' buttons on the right. The configuration fields are: Server (dev10.local:1883), Topic (Metrics/dev10/wlan0/Rx/), QoS (2), and Name (dev10 Rx Rate). The node is marked as 'connected'.

Node-RED

filter nodes

Dashboard

output

- debug
- link
- mqtt
- http response
- websocket
- tcp
- udp
- play audio
- serial

function

dev10 Status abc

dev10 Tx Rate connected

dev10 Rx Rate abc

dev10 Rx Rate

dev10 Tx Rate abc

dev10 Tx Rate

Edit mqtt in node

Delete Cancel Done

Server dev10.local:1883

Topic Metrics/dev10/wlan0/Rx/

QoS 2

Name dev10 Rx Rate

Node-Red Dashboard

The image shows the Node-RED web interface. On the left, the 'output' category is expanded, showing nodes like debug, link, mqtt, http response, websocket, tcp, udp, play audio, and serial. The 'function' category is also visible. The main workspace displays a 'Dashboard' tab with five MQTT nodes: 'dev10 Status', 'dev10 State', 'dev10 Rx Rate', and 'dev10 Tx Rate', all showing a 'connected' status. A fifth node, 'dev 10 Network Activity', is highlighted in the top right of the workspace. An 'Edit chart node' dialog is open for this node, showing configuration options for a line chart.

Node-RED

filter nodes

Dashboard

output

- debug
- link
- mqtt
- http response
- websocket
- tcp
- udp
- play audio
- serial

function

dev10 Status
connected

dev10 State
connected

dev10 Rx Rate
connected

dev10 Tx Rate
connected

dev 10 Network Activity

Edit chart node

Delete Cancel Done

Group dev10 [Network Activity]

Size auto

Label dev10

Type Line chart

X-axis last 1 hours OR 1000 points

X-axis Label HH:mm

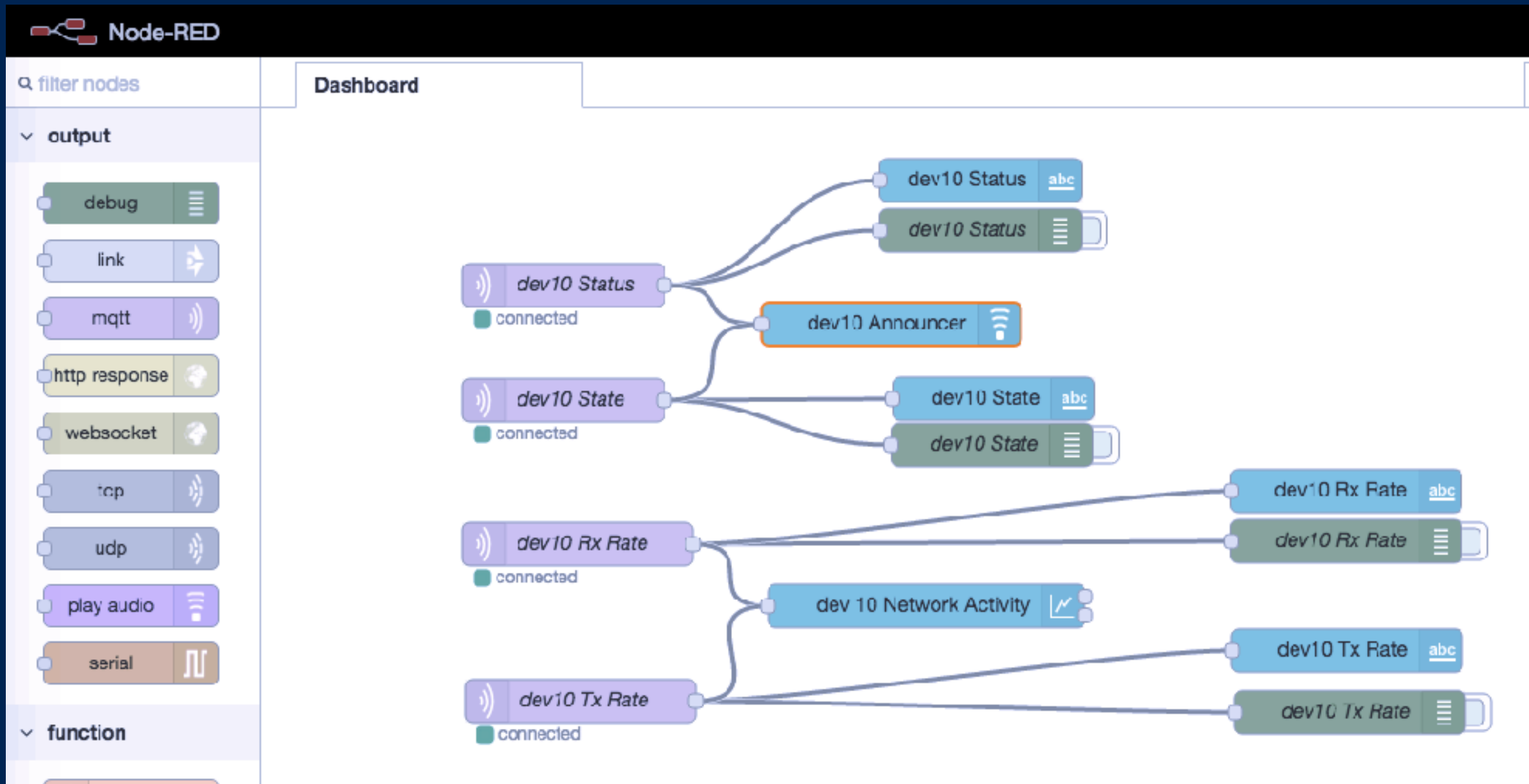
Y-axis min 0 max 5000

Legend Show Interpolate step

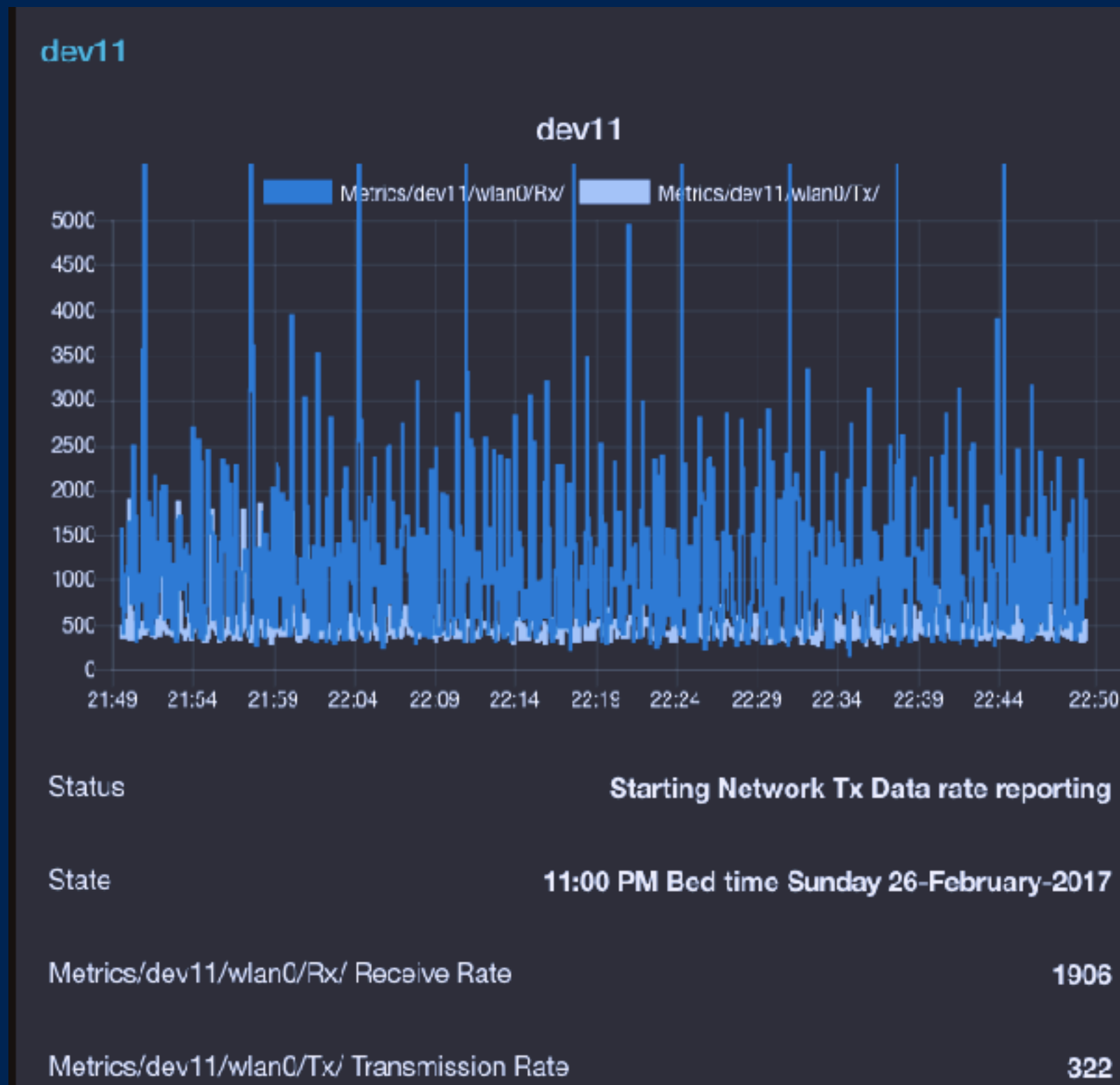
Blank label Waiting for data

Name dev 10 Network Activity

Node-Red Dashboard



Dashboard Output



What comes next?

- ✦ Add code to subscribe to Command topics and act upon commands
- ✦ Add code to applications to send commands to each other
- ✦ Use a Dashboard to take control!

Meanwhile I was side tracked by an Amazon Alexa Voice Skills Workshop

- ✦ That means I have been building voice skills
 - ✦ Tell Siri (working)
e.g. David would like Siri to go buy some chocolate
==> “Hey Siri! Buy some chocolate”
 - ✦ Full Moon (in progress)
e.g. interrogate the US Navy online service for the full moon in a particular month
 - ✦ Mindful Meditations (not started)
e.g. do a relaxing meditation
follow the breath

Meanwhile still planning on TimePeace II

- ✦ Link the AlexaPi to the CalendarTV with NodeRed/MQTT
 - ✦ Send common commands using the CalendarTV remote control to the AlexaPi
 - ✦ Common commands from remote control
“What day is it today?”, “What is happening today?”
 - ✦ Possibly also apply Alexa Skills to use voice via AlexaPi to control CalendarTV & TimePeace using Node-Red
- ✦ An easy to manipulate timeline of personal history & photos in the context of historical events, music, TV, Film

Internet of Things Infrastructure

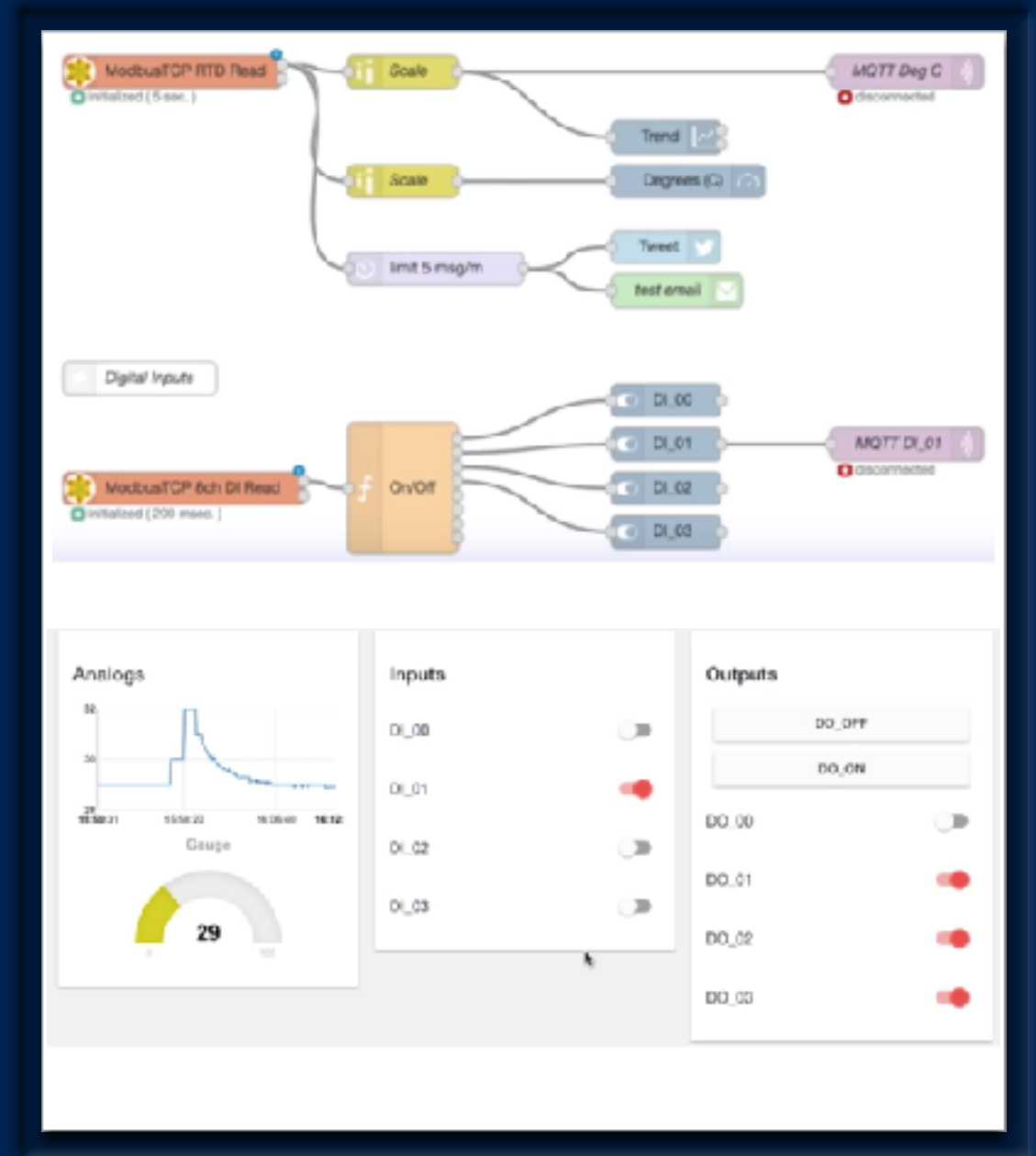
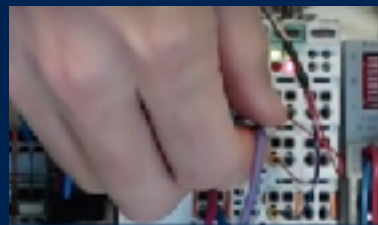
- ✧ Paho MQTT Mosquito

- ✧ Easy to use
- ✧ Publish & Subscribe
- ✧ Fast status & update sharing



- ✧ IBM Node Red

- ✧ Easy to use
- ✧ Part of Raspbian distribution
- ✧ Rapid delivery
- ✧ Graphical orchestration
- ✧ Build status web pages
- ✧ Save data
- ✧ Link systems & coordinate activities



David Penney

email: david.penney@icloud.com

twitter: @david__penney

(note two underscores)



Presentation to “Raspberry Pint London”
29th November 2016 by David Penney

Find us & join in!

Search google for “Raspberry Pint London”
or click on

<https://www.meetup.com/Raspberry-Pint-London/>

<https://www.facebook.com/groups/raspberrypint/>