

# WiFi, Rpi and Lua, Oh My

How to program and communicate with an ESP8266 WiFi device using a Raspberry Pi

# Clone from Github

```
$ cd
```

```
$ git clone \  
http://github.com/Hackerspace-Charlotte/RaspberryPiNight
```

```
$ cd ~/RaspberryPiNight/esp8266/Session2
```

- If luatool is not installed, cd to Session1, get presentation and follow steps.
- Install firefox if not installed:

```
$ sudo apt-get update
```

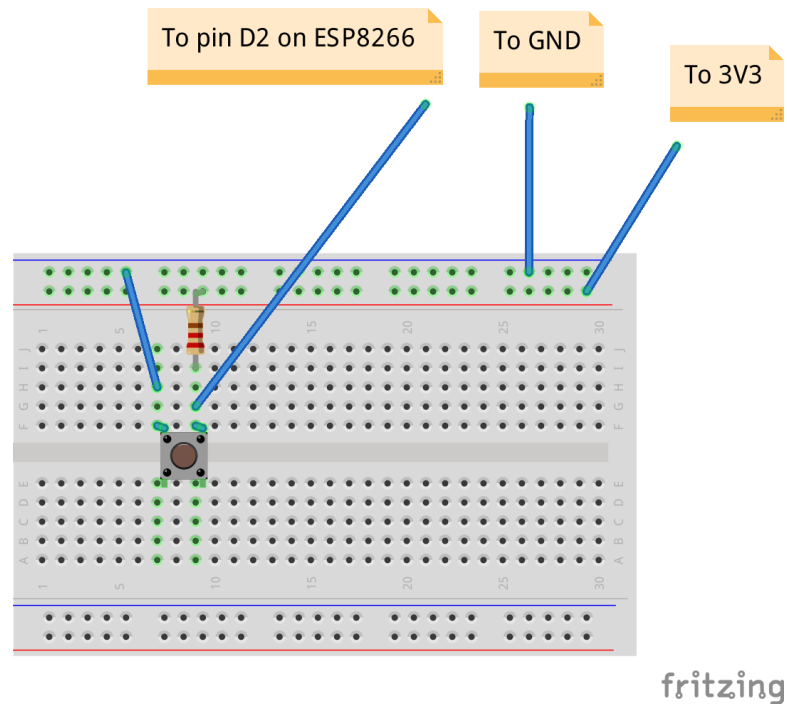
```
$ sudo apt-get install firefox-esr
```

- Write a function to make using luatool easier:

```
$ function luatool() \  
{ luatool.py --port /dev/ttyUSB0 --baud 115200 \  
--src $1 --dest $1; }
```

# Use ESP8266 to detect input

- Create the circuit:



# Use ESP8266 to detect input

- Upload Lua program to detect a switch:

```
$ cd
```

```
~/RaspberryPiNight/esp8266/Session2/switch
```

```
$ luatool init.lua
```

```
$ luatool main.lua
```

- Then connect with picocom:

```
$ picocom --baud 115200 /dev/ttyUSB0
```

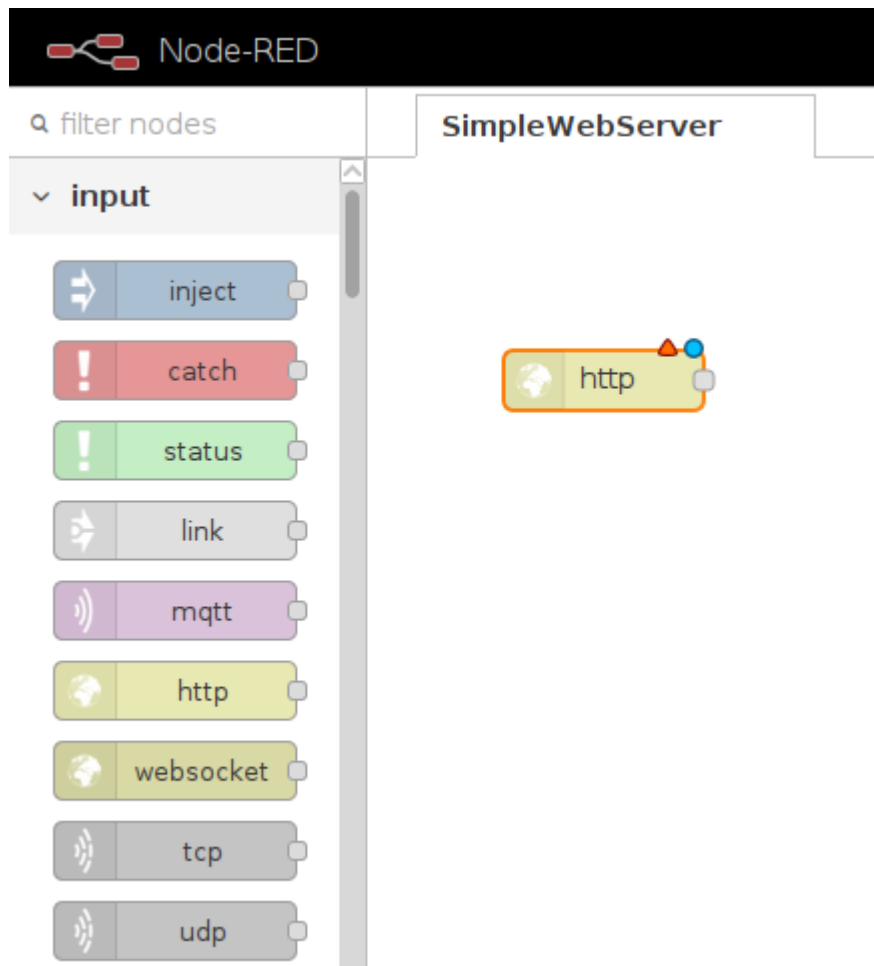
- Press Reset button on ESP8266, take note of IP address that it gets when connected. Will use later.
- Then press and release switch. Output is “Switch pressed” and “Switch released”

# Send the output to a web server

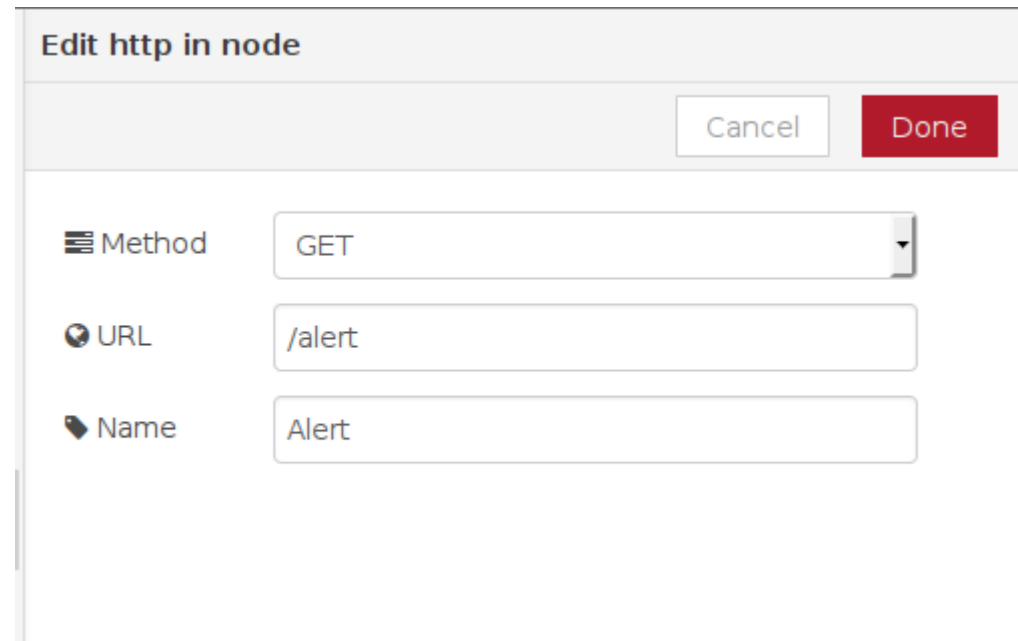
- Use NodeRED to create the web server
- Start NodeRED from the menu
- Open firefox, go to <http://localhost:1880>
- NodeRED interface, ready to build flow.

# Add HTTP input node

Drag “http” node onto flow.



Double click and add URL.



# Add handler node

Drag “function” node onto flow.

Double click and add message.

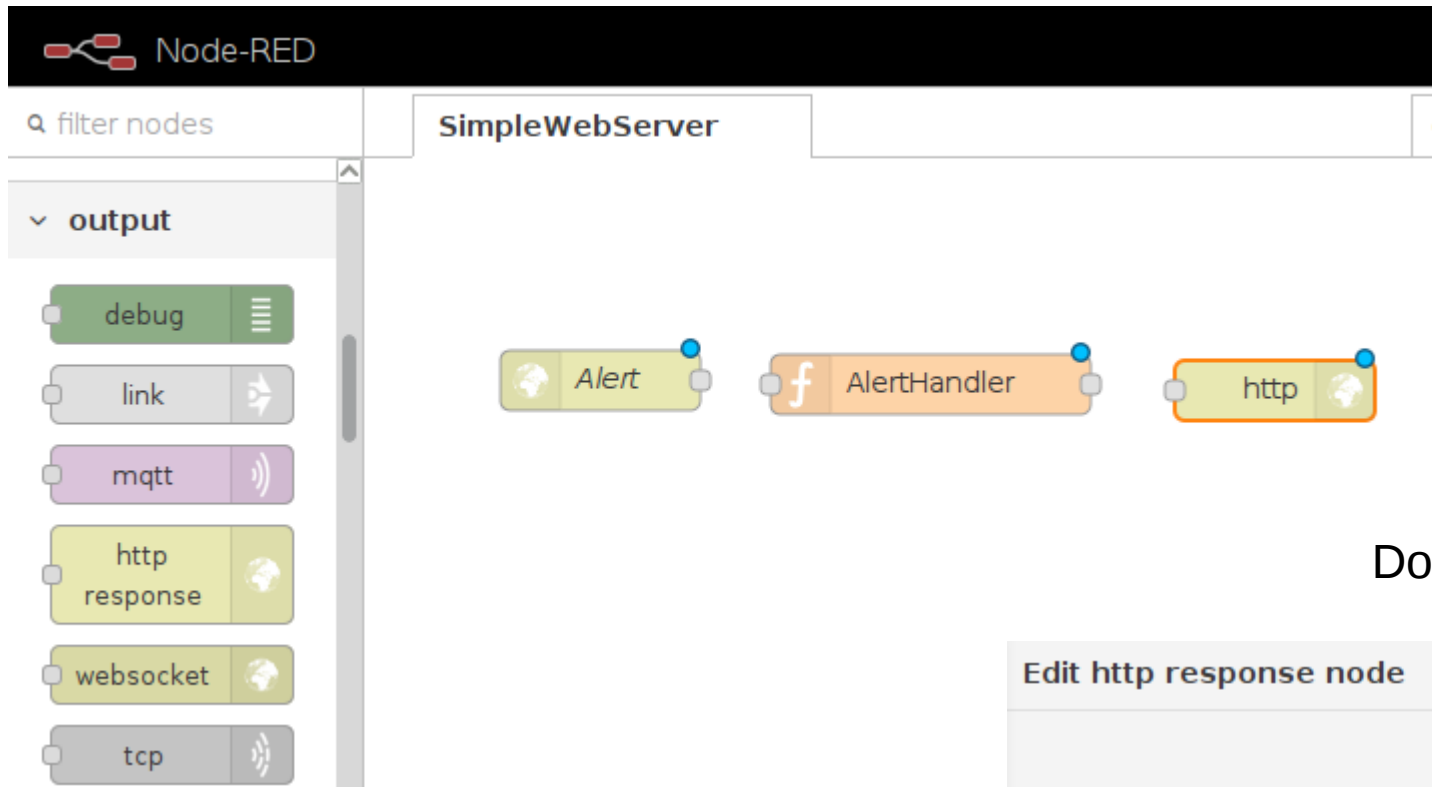
The screenshot shows the Node-RED web interface. On the left, the 'function' category is selected in the node palette, displaying various function-related nodes like 'function', 'template', 'delay', and 'trigger'. In the center workspace, a flow is visible with an 'Alert' node connected to an empty 'function' node. On the right, the 'Edit function node' dialog is open. The 'Name' field is set to 'AlertHandler'. The 'Function' field contains the following JavaScript code:

```
1 msg.payload = "Do something!";  
2 return msg;
```

Below the code editor, the 'Outputs' dropdown is set to '1'. At the bottom of the dialog, a yellow box contains the text: 'See the Info tab for help writing functions.'

# Add HTTP response node

Drag http response node onto flow.



Double click, rename, **deploy**.

Edit http response node

Cancel Done

Name BackAtYa

The messages sent to this node **must** originate from an *http* input node



# Program the ESP8266

- Upload Lua programs:

```
$ cd ~/RaspberryPiNight/esp8266/Session2/sendswitch
```

- Modify main.lua – change `serverurl` to use IP address found earlier.

```
$ luatool init.lua
```

```
$ luatool main.lua
```

- Then connect with picocom:

```
$ picocom --baud 115200 /dev/ttyUSB0
```

- Press Reset button
- Press switch until it says “Switch pressed”. Output should show HTTP request and response after switch is pressed.

# Add email alert node

Drag email *output* node onto flow.

The screenshot shows the Node-RED web interface. At the top, the title bar says "Node-RED". Below it, a search bar contains "filter nodes". A sidebar on the left lists various nodes: "json", "xml", "rbe", "random", "smooth", and a "social" category. Under "social", there are two "e mail" nodes (green with an envelope icon) and two "twitter" nodes (blue with a bird icon). The main workspace shows a flow with three nodes: "Alert" (green with a globe icon), "AlertHandler" (orange with a function icon), and "BackAtYa" (green with a globe icon). A new "EmailAlert" node (green with an envelope icon) is being dragged from the sidebar into the flow, connected to the "AlertHandler" node.

Double click and edit.

The "Edit e-mail node" dialog box is open, showing configuration fields for an email alert. The fields are: "To" (you@somewhere.com), "Server" (smtp.gmail.com), "Port" (465), "Userid" (nlrrpi@gmail.com), "Password" (masked with dots), and "Name" (EmailAlert). There are "Cancel" and "Done" buttons at the top right.

Edit e-mail node	
To	you@somewhere.com
Server	smtp.gmail.com
Port	465
Userid	nlrrpi@gmail.com
Password	.....
Name	EmailAlert

# Check email

- Connect with picocom

```
$ picocom --baud 115200 /dev/ttyUSB0
```

- Press Reset button, verify it says “Switch released”
- Press switch, watch flow.
- Flow should say “Sending” briefly.
- Check your email.