Raspberry Pi Deployment

Matthias Dieter Wallnöfer, TIS innovation park, Bolzano/Bozen - Italy

- Node-RED
  - Preparation - Installation of Node-RED
  - Make Node-RED startable on boot
  - Use Node-Red to deploy all flows to all the Pis
- WLAN configuration
- Image backup

Node-RED

Preparation - Installation of Node-RED

- Installation and configuration of RaspBian
- Copy these 2 files onto the Pi: node-red-0.8.1.zip and node-red-nodes-master.zip (the nodes)
- Perform these 2 commands in order to install node.js:
  wget http://node-arm.herokuapp.com/node_latest_armhf.deb
  sudo dpkg -i node_latest_armhf.deb
- Install serial port node.js module if needed (eg. for Arduino)
  npm install serialport
- In the home directory
  unzip node-red-0.8.1.zip
- in the node directory under the Node-RED directory
  unzip node-red-nodes-master.zip
- In the Node-RED directory
  npm install --production
- Create a Node-RED starter script in the home directory and make it executable (chmod +x) noderedstarter.sh:
  cd /home/pi/node-red-0.8.1
  node --max-old-space-size=128 red.js
- Install eventual other scripts and programs used by the Node-RED flows
- Launch Node-RED in pi's homedirectory:
  ./noderedstarter.sh
- Deployment:
  curl -X POST -i -H "Content-type: application/json" -d @<pi flow filename eg. node-red-0.8.1/flows_pi1.json> http://<pi_address>:1880/flows

Make Node-RED start at boot

- Modify */etc/rc.local* to add this line before *exit 0;*
  sudo -u pi /home/pi/noderedstarter.sh &

- and the *noderedstarter.sh* should me modified to track events for debugging:
  [...] > /home/pi/nodered.log 2> /home/pi/nodered.err.log

Use Node-RED to deploy all flows to all the Pis ("pi-deployment")

There are 2 flows:

1. The real application that will be deployed on each Pi, eg. to read the temperature.
2. Simple deployment flow that takes in input the Pi flow and delivers it to all Pis using curl.

How-to

- The PI flow:
  - *flows_raspberrypi.json*: the Pi Node-RED flow which should be deployed, equal for each Pi
- The deployment flows:
  - *flows_deploy.json*: this is a very basic deployment flow, useful for a single Pi deployment
  - *flows_deploy2.json* this is thought for massive-deployment
- Copy the preferred deployment flow file into the **host's** Node-RED directory with filename *flows_<hostname>.json*
- Restart/start **host's** Node-RED (*node red.js* in Node-RED's directory)
- Adjust the filenames in the file components using the webbrowser (*http://localhost:1883*). This has to be done once.
- For the simple flow, adjust the Pi address in the HTTP request component
- For the complex flow edit the file *pi_addresses.csv*. Syntax is *<address>,<description>*, # is the comment sign
- Click the red "Deploy" button in Node-RED's web interface's top right corner
- Make sure that all Pi Node-RED installations are running, otherwise you get errors back
- Now press the blue "Inject" button in the flow, deployment should happen
- HTTP status *204* is showed up for each successfully deployed Pi in the "Debug" tab

NB: the procedure has been tested on two Pis with no problems.

WLAN configuration

This is a short how-to to make a WLAN stick on the Pi work.

- Connect WLAN stick to USB and (re)boot
- Run as root:
  wpa_passphrase <SID> >> /etc/wpa_supplicant/wpa_supplicant.conf
- Type in WLAN password
- Reboot or restart WLAN interface
  sudo ifdown wlan0
  sudo ifup wlan0
- Modify the WLAN section of /etc/network/interfaces to look like this:
  allow-hotplug wlan0
  iface wlan0 inet dhcp

wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
iface default inet dhcp

Image backup

Use a command like this:
sudo dd if=/dev/mmcblk0 of=pi2_paolo.img bs=4M
Restore*:*
sudo dd if=pi2_paolo.img of=/dev/mmcblk0