Raspberry Pi Survival Guide

The following only needs to be done once. If you are working on the Goddard Project, I have already done this, else, after completing these 10 steps you will be able to control the Raspberry Pi remotely.

1. Connect HDMI, keyboard, networking (wired or open wireless), power
2. Username: pi Password: raspberry
3. Ensure SSH is enabled
4. Install avahi-daemon (http://elinux.org/RPi_Advanced_Setup)
    - sudo apt-get install avahi-daemon
    - sudo nano /etc/avahi/services/multiple.service
    - paste or type the following:

```
<?xml version="1.0" standalone='no'?>
<!DOCTYPE service-group SYSTEM "avahi-service.dtd">
<service-group>
<name replace-wildcards="yes">%h</name>
      <service>
            <type>_device-info._tcp</type>
            <port>0</port>
            <txt-record>model=RackMac</txt-record>
      </service>
      <service>
            <port>22</port>
      </service>
</service-group>
```

    - sudo /etc/init.d/avahi-daemon restart
    - Windows Users need to follow step 4
5. Congratulations! Your Raspberry Pi can now be used “headless”.
6. logout
7. Unplug keyboard/HDMI
8. SSH Into Raspberry Pi @ raspberrypi.local port 22
    - Windows users need PuTTY
    - Mac/Linux open terminal and type: ssh raspberrypi.local
    - If the host is unreachable, try the Pi's ip address instead
9. sudo apt-get update
10. sudo apt-get upgrade

Now that the Pi is updated and SSH has been configured lets turn it into a server! If you are in the Goddard group I have already done this.

1. Install Node.JS (http://blog.rueedlinger.ch/2013/03/raspberry-pi-andnodejs-basic-setup/)
   - sudo mkdir /opt/node
   - wget http://nodejs.org/dist/v0.10.2/node-v0.10.2-linux-arm-pi.tar.gz
   - tar xvzf node-v0.10.2-linux-arm-pi.tar.gz
   - sudo cp -r node-v0.10.2-linux-arm-pi/* /opt/node
   - nano /etc/profile
   - add the following lines just before “export path”

```
NODE_JS_HOME="/opt/node"
PATH="$PATH:$NODE_JS_HOME/bin"
export PATH
```

   - logout

2. Configure NodeJS (http://blog.rueedlinger.ch/2013/03/raspberry-pi-and-nodejs-basic-setup/)
   - cd ~
   - nano nodejs.sh
     - add the following:

```
#!/bin/bash
NODE=/opt/node/bin/node
SERVER_JS_FILE=/home/pi/app/server.js
USER=pi
OUT=/home/pi/nodejs.log
case "$1" in
start)
      echo "starting node: $NODE $SERVER_JS_FILE"
      sudo -u $USER $NODE $SERVER_JS_FILE > $OUT 2>$OUT
      &
      ;;
stop)
      killall $NODE
      ;;
*)
      echo "usage: $0 (start|stop)"
esac
exit 0
```

   - Make the shell script executable
     - chmod 755 nodejs.sh
     - sudo cp nodejs.sh /etc/init.d
     - sudo update-rc.d nodejs.sh defaults
   - Make a folder for a new NodeJS app
     - mkdir /home/pi/app

3. Implement a Server in NodeJS (http://blog.rueedlinger.ch/2013/03/raspberry-pi-and-nodejs-basic-setup/)
    - cd /home/pi/app
    - nano server.js
        - add the following:

```
var http = require('http');

http.createServer(function(req,resp) {
      resp.writeHead(200, {"Content-Type":
      "text/plain"});
      resp.write("Hello World");
      resp.end();

      console.log("sample output to console");
}).listen(8080);
```

    - Start the NodeJS server
        - node server.js
    - Open a web browser and type raspberrypi.local:8080
    - If you see “Hello World!” than you did it! If you don't, try using <raspberrypi's ip address>:8080, if this works then you failed to install avahi-daemon.

You now have a bare-bones web server. RaspberryPiSurvival_Part2 will teach advanced skills like configuring a web application framework and implementing an application that talks to GPIO.

@bobbyg603

Appendix:

stop node	killall node
check ip	ifconfig
renew ip	sudo dhclient <wlan0 or eth0>
compiling	chmod 755 <filename>
move file	mv <source> <full path to destination>
copy file	cp <source> <full path to destination>
log out	logout
log out another user	pkill -KILL -u <username>

Update NodeJS (replace xx with minor version number):

- sudo rm /opt/node -r
- sudo mkdir /opt/node
- wget http://nodejs.org/dist/v0.10.xx/node-v0.10.xx-linux-arm-pi.tar.gz
- tar xvzf node-v0.10.xx-linux-arm-pi.tar.gz
- sudo cp -r node-v0.10.xx-linux-arm-pi/* /opt/node