JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY
1990

Raspberry Pi 3 Getting Started Guide

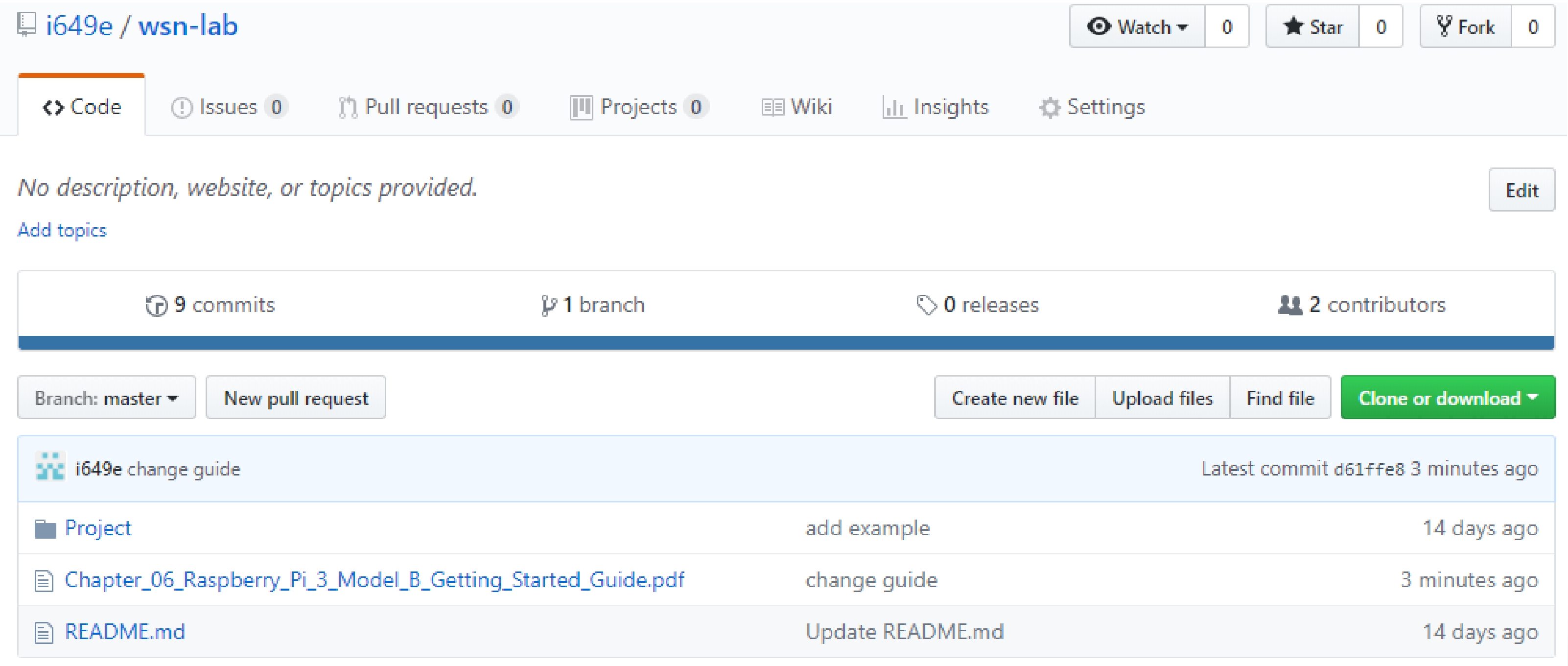
Nantawat Pinkam

I649E Wireless Sensor Networks

Update June 28, 2018

I649E's Github

https://github.com/i649e/wsn-lab



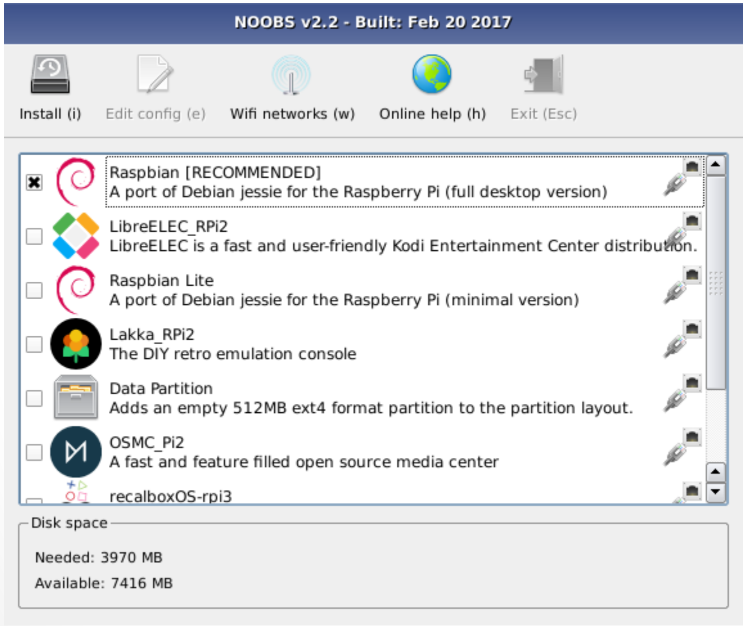
No description, website, or topics provided.

9 commits | 1 branch | 0 releases | 2 contributors

i649e change guide — Latest commit d61ffe8 3 minutes ago

Project	add example	14 days ago
Chapter_06_Raspberry_Pi_3_Model_B_Getting_Started_Guide.pdf	change guide	3 minutes ago
README.md	Update README.md	14 days ago

Installation (NOOBS) with monitor+mouse+keyboard



☒ Raspbian [RECOMMENDED]
A port of Debian jessie for the Raspberry Pi (full desktop version)

□ LibreELEC_RPi2
LibreELEC is a fast and user-friendly Kodi Entertainment Center distribution.

□ Raspbian Lite
A port of Debian jessie for the Raspberry Pi (minimal version)

□ Lakka_RPi2
The DIY retro emulation console

□ Data Partition
Adds an empty 512MB ext4 format partition to the partition layout.

□ OSMC_Pi2
A fast and feature filled open source media center

recalboxOS-rpi3

Disk space

Needed: 3970 MB

Available: 7416 MB

1. Download NOOB's Raspberry Pi OS from

   https://www.raspberrypi.org/downloads/noobs/

or from mirror at JAIST

   http://ftp.jaist.ac.jp/pub/raspberrypi/NOOBS/images/

2. Extract files into your SD card
3. Put the SD card back to Raspberry Pi
4. Boot and install (first item). It takes around 30 minutes to complete.

Installation (Etcher) without monitor+mouse+keyboard

1. Download Raspbian Desktop at:

http://ftp.jaist.ac.jp/pub/raspberrypi/raspbian/images/raspbian-2018-04-19/2018-04-18-raspbian-stretch.zip

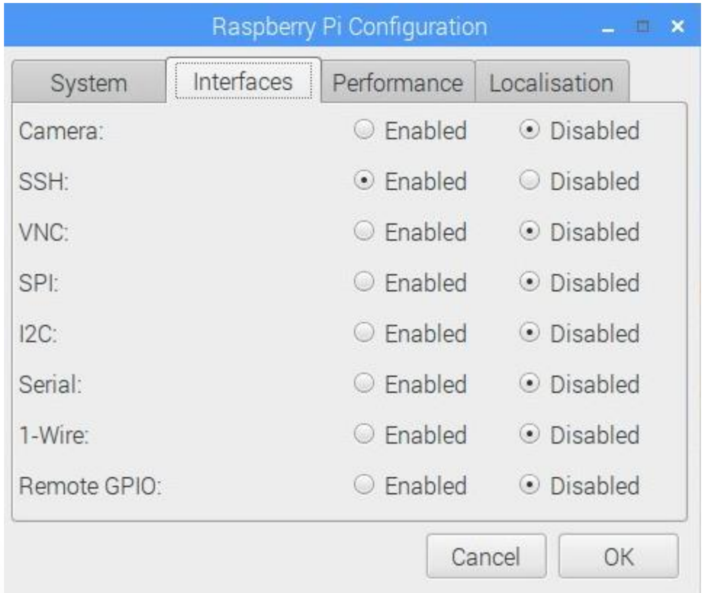
2. Extract file into your computer.
3. Download Etcher at:

https://etcher.io

4. Insert SD card and run Etcher
5. Select Raspbian image file and SD card's drive then click Flash!
6. After finish, **unplug** the SD card, wait for a while and **put it back**.
7. Create file name **ssh** in the **root folder** (without extension)
8. Create file name **wpa_supplicant.conf** in the **root folder** and edit this file as:

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
network={
    ssid="wsn-lab"
    psk="i649ewsn"
    key_mgmt=WPA-PSK
  }
```

Connect Raspberry Pi through SSH (with monitor+mouse+keyboard)



Camera:	○ Enabled	● Disabled
SSH:	● Enabled	○ Disabled
VNC:	○ Enabled	● Disabled
SPI:	○ Enabled	● Disabled
I2C:	○ Enabled	● Disabled
Serial:	○ Enabled	● Disabled
1-Wire:	○ Enabled	● Disabled
Remote GPIO:	○ Enabled	● Disabled

i. Start > Preferences > Raspberry Pi Configuration

ii. Set Hostname e.g. node3

iii. Interface tab > Enable SSH

iv. You can enable VNC for remote desktop capability though VNC Viewer

v. Use this for login:

User:	pi
Password:	raspberry

Connect Raspberry Pi through SSH

- *Windows*
  1. Download PuTTY and install from https://putty.org
  2. Enter Hostname.local or IP address

```
node5.local
10.10.0.5
```

- *Mac and Linux*
  1. Open terminal
  2. Type

```
ssh pi@<IP address>
ssh pi@<Hostname>
ssh pi@10.10.0.4
ssh pi@node4.local
```

Use this for login:
User: pi
Password: raspberry

Change Hostname

1. Connect ssh to raspberry pi
2. Use command:

```
sudo raspi-config
```

3. Navigate to **Network Options**.
4. Select **N1 Hostname**
5. Change your Pi’s Hostname to **nodexxx** (x is the number of your Pi e.g. node001, node015)
6. Choose < Finish >

Connect Raspberry Pi through VNC (Command Line)

1. Connect ssh to raspberry pi
2. Use command:

```
sudo raspi-config
```

3. Navigate to **Interfacing Options**.
4. Scroll down and select **VNC > Yes**.
5. Download **VNCViewer**

https://www.realvnc.com/en/connect/download/viewer/

6. Input **Hostname.local** or **IP address**

Update and Install Essential Software

```
sudo apt-get update -y
sudo apt-get upgrade -y
sudo apt-get dist-upgrade -y
sudo pip install --upgrade pip
sudo apt-get install olsrd iperf wavemon python-numpy python-scipy python-matplotlib -y
```

Raspberry Pi wlan0 Ad-Hoc Setup

- Edit the interfaces file in /etc/network/

```
sudo nano /etc/network/interfaces
```

- Add the following

```
auto wlan0
iface wlan0 inet static
    address <IP address>
    netmask 255.255.255.0
    mtu 1500
    wireless-channel <channel>
    wireless-essid <network name>
    wireless-mode ad-hoc
    wireless-ap any
```

- Reboot Raspberry Pi to take effect.

Setup ad-hoc Network in macOS

1. Open System Preferences and select “Network”. Choose Wi-Fi on the left and enable “Show Wi-Fi status in menu bar”.
2. Click on Wi-Fi status on the top right menu bar.
3. Click create network.
4. Input network name of your raspberry pi and channel
5. Back to Network in System Preferences, click Advance…
6. In TCP/IP tab, Choose Manual in Configure IPv4
7. Input IPv4 Address 10.10.0.x and subnet mask 255.255.255.0

Setup ad-hoc Network in Windows (EXCEPT SURFACE)

1. Run command prompt as administrator
2. Type

```
netsh wlan set hostednetwork mode=allow
ssid=<network name> key=
```

3. Then type

```
netsh wlan starthostednetwork
```

4. Set your network IP address in network connection properties

Optimized Link State Routing Protocol Daemon (OLSRD)

- Start OLSRD using wlan0 interface on Raspberry Pi with debug level 1

```
sudo olsrd -i wlan0 -d 1
```

- Check the communication by using ping command

```
ping <IP address>
```

- For multi-hop capability. You need to force two nodes to use a gateway by using firewall to block each other. There is no direct implementation on OLSRD.

```
sudo iptables -A INPUT -m mac --mac-source XX:XX:XX:XX:XX:XX -j DROP
```

- Use the command route or traceroute to see if they use a gateway

```
route
traceroute <IP address>
```

Throughput Measurement

The command is `iperf`. `iperf` is a tool for active measurements of the maximum achievable bandwidth on IP networks.

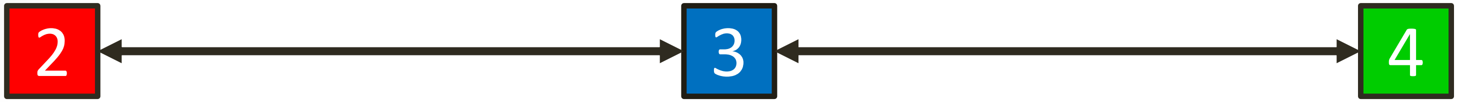
- Server side

```
sudo iperf -s
```

- Client side

```
sudo iperf -c <IP address> -t <time in second>
```

Test 2-hop network



1. Run olsrd on all devices

```
sudo olsrd -i wlan0 -d 1
```

2. Force **node002** to use **node003** as a gateway to **node004**

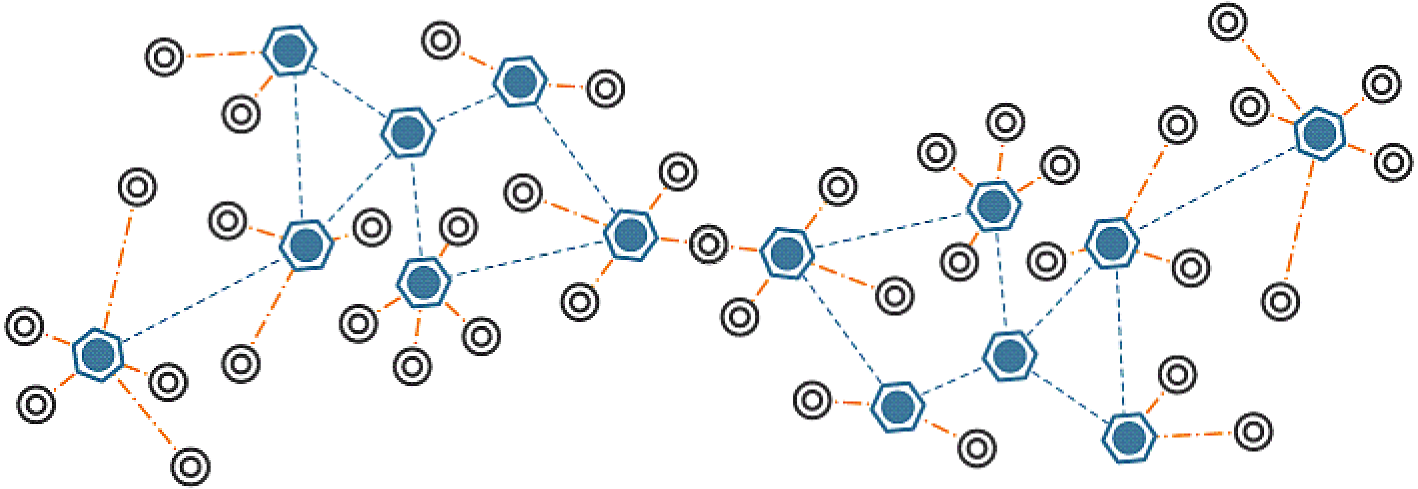
```
sudo iptables -A INPUT -m mac --mac-source <node004 mac address> -j DROP
```

3. Force **node004** to use **node003** as a gateway to **node002**

```
sudo iptables -A INPUT -m mac --mac-source <node002 mac address> -j DROP
```

Test	node002	node003	node004
ping	ping 10.10.0.4	ping 10.10.0.2 ping 10.10.0.4	ping 10.10.0.2
traceroute	traceroute 10.10.0.4	traceroute 10.10.0.2 traceroute 10.10.0.4	traceroute 10.10.0.2
netcat	netcat -l -p 55555		netcat 10.10.0.2 55555
iperf	sudo iperf -s	sudo iperf -s	sudo iperf -c 10.10.0.2 -t 10 sudo iperf -c 10.10.0.3 -t 10

Question and Answer



THANK YOU.