Formation of wireless sensor network using 6lbr (6LoWPAN Border Router) and TI cc2650stk sensortag.

By following the steps present in the document, one can easily create a wireless sensor network based on 6LoWPAN. There are 4 major objectives to do so.

1. Installing Contiki-NG in the Linux Host and configuring it
2. Installing and Booting Raspberry Pi
3. Building Slip Radio with Launchpad CC2650
4. Building web-demo example for the end nodes (sensortag)
5. Building 6lbr on the RaspBerry Pi
6. Integrating the setup as desired with the above components
7. Building Slip Radio with Launchpad CC2650

Before achieving any objectives the following hardware and softwares will be required:

## Software Requirements:

* 1. [Uniflash](https://www.ti.com/tool/UNIFLASH)

For flashing the files to the devices. Can also use SmartRFProgrammer 2 (Available in Drive)

* 1. [Rpi-Imager](https://github.com/raspberrypi/rpi-imager#:~:text=Download%20the%20latest%20version%20for,sudo%20apt%20install%20rpi%2Dimager%20.)

For installing OS on RaspberryPi

* 1. [Angry IP Scanner](https://github.com/angryip/ipscan/releases/download/3.8.2/ipscan_3.8.2_all.deb)

To scan and find out the devices present in the same network

* 1. [Putty](https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html)

For making ssh connection to the headless RaspberryPi

* 1. [Configuration Files](https://drive.google.com/file/d/1ovA5uCcsZydC5x5IdY56M77JXE_gBsWR/view?usp=sharing)

Download/View from Google Drive: 6lbr.conf, interfaces, sysctl (2 lines to uncomment)

## Hardware Requirements:

1. A Linux Host Machine (Ubuntu 18.04 LTS)
2. RaspberryPi 4
3. 32/64 GB MicroSD card and card reader
4. RaspberryPi Official Power Adapter (3A)
5. TI SensorTag cc2650stk + Micro USB Cable
6. TI Launchpad CC2650 + MicroUSB Cable
7. Ethernet cable (to make serial connection between Raspberry Pi and host machine)
8. Wireless Router\*

\*In case, the host system is a desktop, then use a WiFi adapter to connect to the wireless router.

**Installing Contiki-NG in the Linux Host and configuring it**

Install Uniflash, RPi and Putty in the Linux host system.

Go to the WIred Network settings and click on the “+” button on the up-right hand side.

Click on the IPv4 tab of the new profile and choose the “Shared to other computers” options from the available IPv4 method.

save it with a new name in the identity tab

cd /etc/X11

sudo nano Xwrapper.config

change the last two lines of the text as follows:

allowed\_users=anybody

needs\_roots\_rights=yes

Install contiki-NG in the system by following the steps given below:

* 1. sudo apt install build-essential doxygen git curl wireshark python-serial srecord rlwrap autoconf automake libxmu-dev gcc-msp430 openjdk-8-jdk ant
  2. wget <https://armkeil.blob.core.windows.net/developer/Files/downloads/gnu-rm/9-2020q2/gcc-arm-none-eabi-9-2020-q2-update-x86_64-linux.tar.bz2>
  3. tar -xjf gcc-arm-none-eabi-9-2020-q2-update-x86\_64-linux.tar.bz2
  4. gedit /home/pradeepkumar/.bashrc
  5. Edit the following line:

export PATH=$PATH:/home/pradeepkumar/gcc-arm-none-eabi-5\_2-2015q4-20151219-linux/bin

* 1. update-alternatives --config java
  2. gedit /home/pradeepkumar/.bashrc
  3. Edit the following line:

export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64/jre

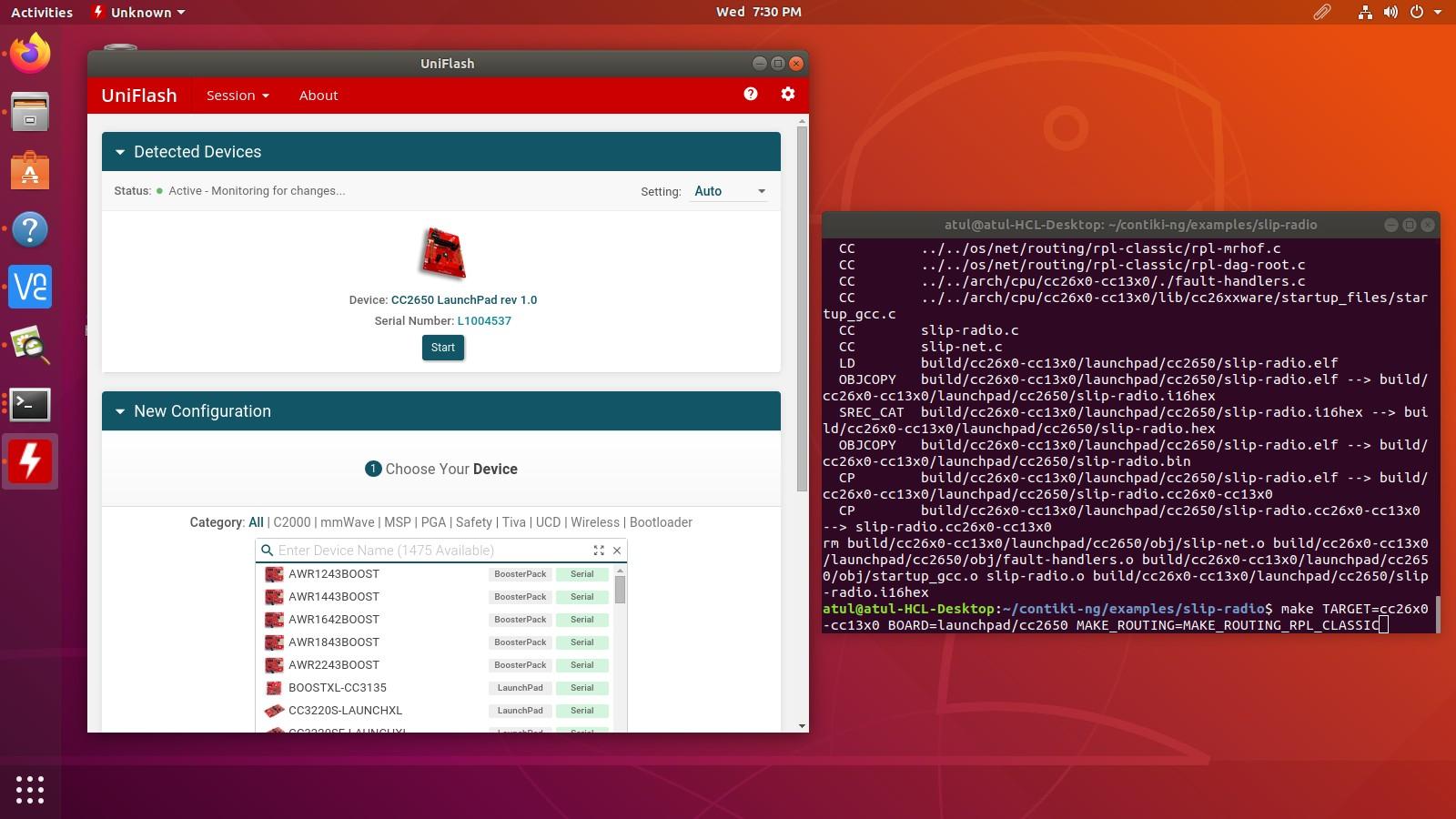
* 1. cd
  2. git clonehttps://github.com/contiki-ng/contiki-ng
  3. cd contiki-ng
  4. git submodule update --init --recursive

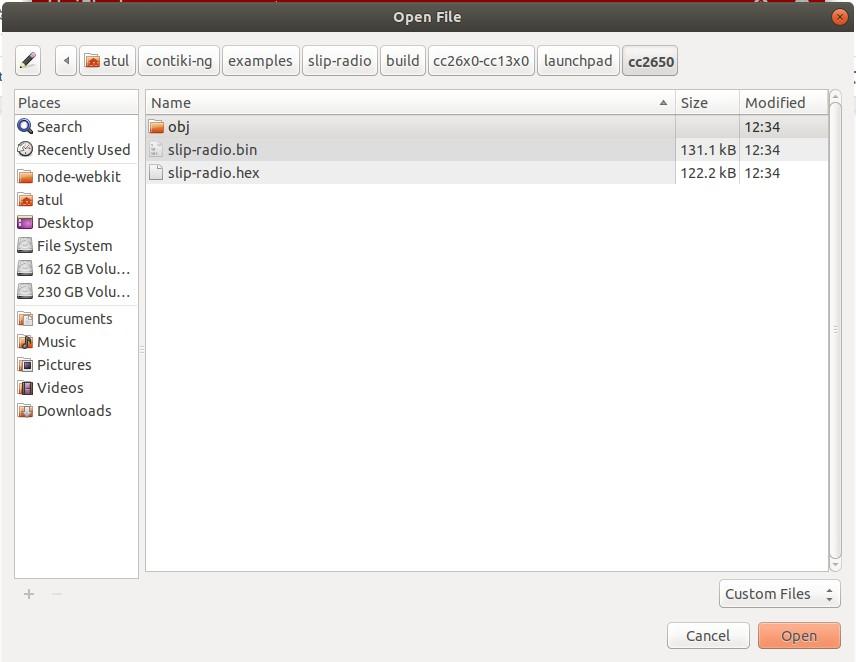
**In order to install RPi image and boot the Raspberry Pi 4 do the following step:**

1. Connect the Micro-SD memory card (32/64 GB) with card reader to the linux host system.
2. Format the SD card and make the file system FAT32 (if required)
3. Burn the SD card with RPi image by selecting the “Raspberry Pi OS (Legacy) with security updates and desktop environment”
4. Flash it
5. After completion remove the SD card from the card reader and connect it to the Raspberry Pi
6. connect the one side of the ethernet cable to the Linux host and other side to the Raspberry Pi
7. Choose the newly created network from the network settings
8. Open Angry IP scanner to find out the IP address assigned to Raspberry Pi [usually it should be 10.42.0.(2-254)]
9. Open putty and type the IP address found under the Host Name
10. click the ‘>’ beside SSH of the left panel and then click on the X11
11. Tick the “Enable X11 Forwarding”
12. Now click on the “open” button below
13. Putty will open a dialog box and clck on the Accept
14. Login ID: pi
15. Password: raspberry
16. sudo apt-get raspi-config
17. go to the interface-options and enable SSH and vnc-server using up-down arrow and tab
18. sudo apt-get update
19. sudo apt-get install tightvncserver
20. sudo apt-get install xtightvncviewer
21. sudo reboot
22. follow the steps h to o
23. vncserver (A new 8 character password will be created)
24. xtightvncviewer -x11cursor
25. server: “raspberrypi:1”
26. password: <8 character password>

**Performed on a Linux Host Machine (Ubuntu 18.04 LTS)**

# Building Slip Radio for Launchpad CC2650

* 1. Install Contiki-NG as per the [Installation Guide](https://docs.google.com/document/d/1t8srGmjgn1cmg5t34IHkf9-JDIRpUOUHqfY4BCY0s24/edit?usp=sharing)
  2. cd contiki-ng/examples/slip-radio
  3. make TARGET=cc26x0-cc13x0 BOARD=launchpad/cc2650 MAKE\_ROUTING=MAKE\_ROUTING\_RPL\_CLASSIC
  4. If getting error for arm use: sudo apt install gcc-arm-none-eabi and repeat step 3
  5. Connect launchpad via USB to the host machine
  6. Open Uniflash
  7. Launchpad will get detected
  8. Browse location of the files created in step 3 and select the bin file to flash



* 1. Click on “Load Image” then “Verify Image”
  2. Thus the Launchpad CC2650 is ready to act as a Slip Radio

# Building Web-Demo example for End Nodes

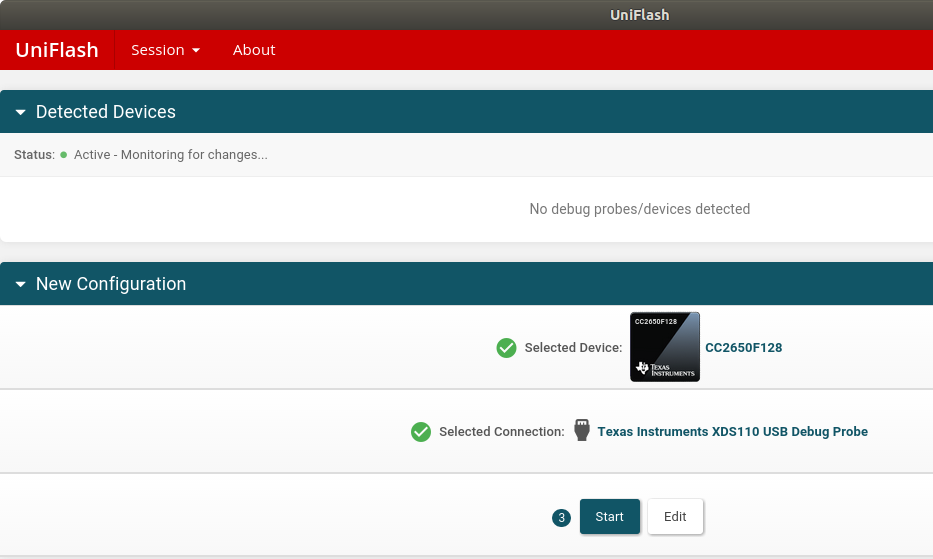
* + 1. cd contiki-ng/examples/platform-specific/cc26x0-cc13x0/cc26x0-web-demo
    2. Set BLE=0 and Default Channel = 25 in project-conf.h file in the above location
    3. If building for SensorTag cc2650stk

make TARGET=cc26x0-cc13x0 BOARD=sensortag/cc2650 MAKE\_ROUTING=MAKE\_ROUTING\_RPL\_CLASSIC

* + 1. If building for Launchpad CC2650

make TARGET=cc26x0-cc13x0 BOARD=launchpad/cc2650 MAKE\_ROUTING=MAKE\_ROUTING\_RPL\_CLASSIC

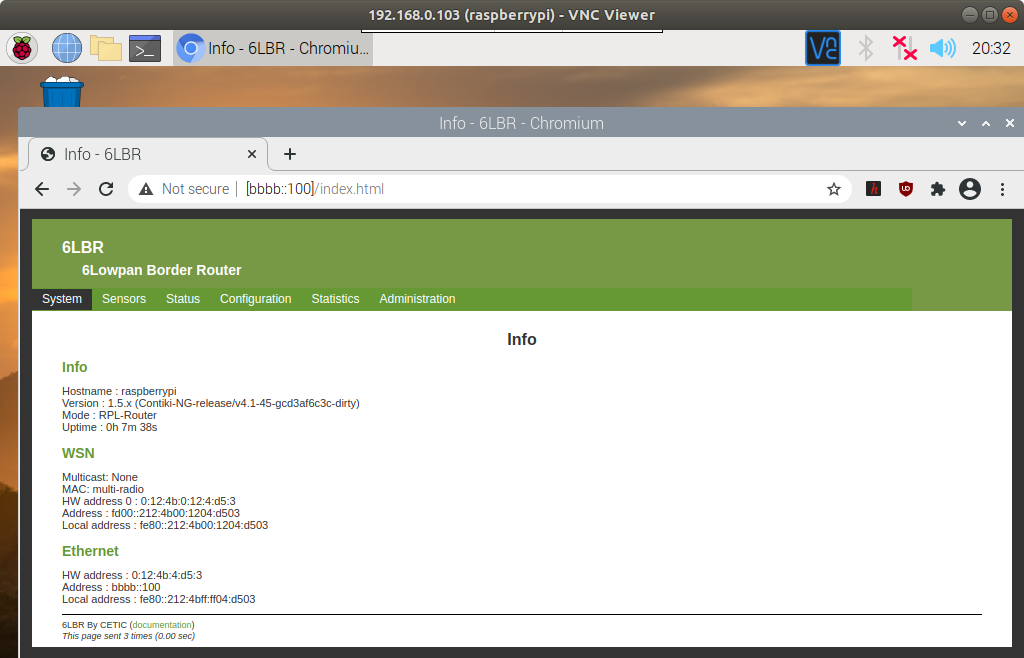
* + 1. Connect launchpad/sensortag via USB to the host machine
    2. Open Uniflash
    3. Launchpad will get detected
    4. For Sensortag use CC2650F128 with XDS110 USB Debug Probe



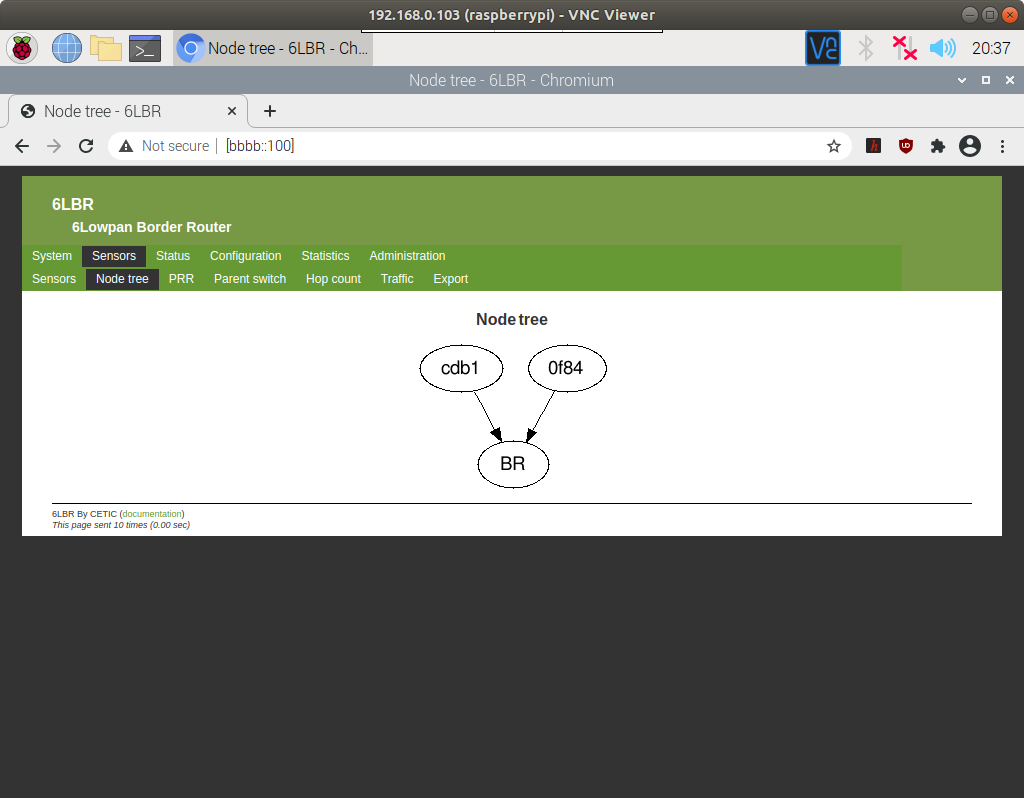
* + 1. Browse location of the files created in step b/c and select the bin file to flash
    2. Load -> Verify Image and your Launchpad/Sensortag is ready to be used as an End Node.

# Building 6lbr on the RaspBerry Pi

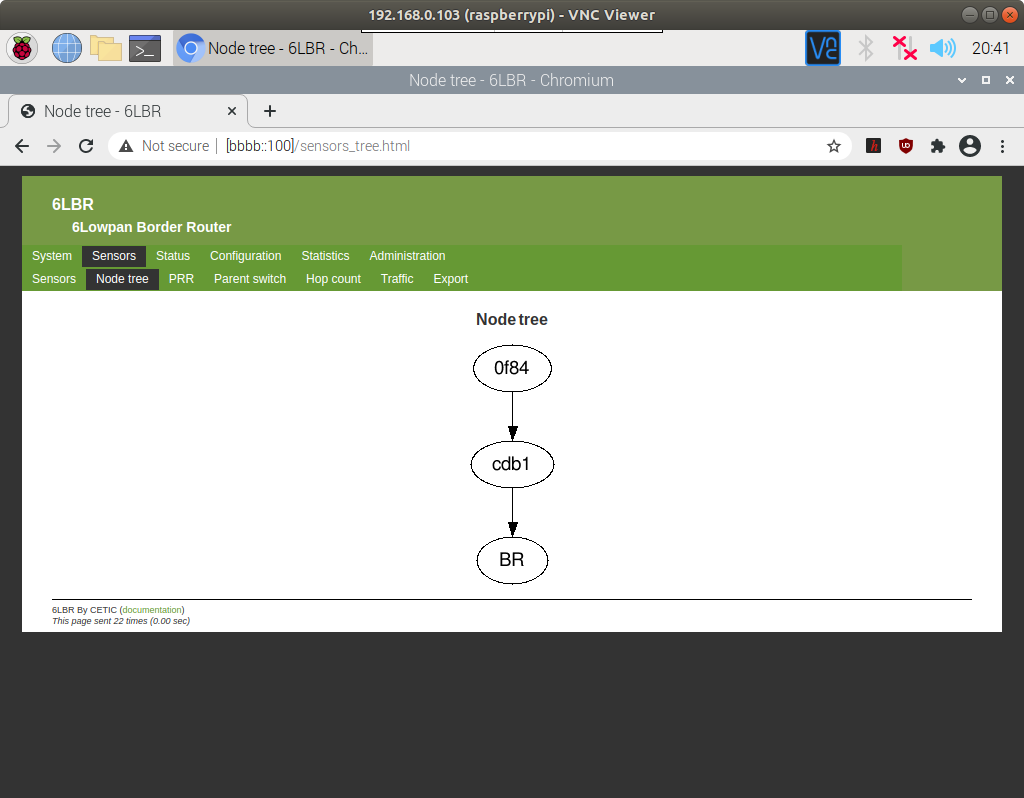
1. SSH login to Raspberry Pi.
2. sudo apt-get install libncurses5-dev
3. sudo apt-get install bridge-utils
4. git clone --recursive https://github.com/cetic/6lbr.git
5. cd 6lbr
6. git submodule update --init
7. cd
8. git clone --recursive https://github.com/cetic/contiki-ng.git
9. cd contiki-ng
10. git submodule update --init
11. cd
12. cd 6lbr/examples/6lbr
13. make WITH\_CONTIKI=0 WERROR=0 all
14. sudo make plugins
15. sudo make tools
16. sudo make install
17. sudo make plugins-install
18. update-rc.d 6lbr defaults
19. sudo /usr/lib/6lbr/bin/nvm\_tool --update --channel 25 /etc/6lbr/nvm.dat
20. /usr/lib/6lbr/bin/nvm\_tool --print /etc/6lbr/nvm.dat
21. cd
22. copy and paste 6lbr.conf from the downloaded configuration files to /etc/6lbr/ location
23. sudo nano /etc/network/interfaces to make the changes mentioned in the downloaded file
24. sudo nano /etc/sysctl.conf to make the changes mentioned in the downloaded file
25. sudo reboot
26. ifconfig or ip a (check if contents are similar or not)
27. sudo service 6lbr start
28. http://[bbbb::100]/index.html



* Single Hop Network:



* MultiHop Network



# Building a service to start 6lbr service at boot.

No display will be required further to start the 6lbr at Raspberry pi manually. The service will start at boot,

1. open a new terminal in raspberry pi
2. sudo nano 6lbr\_start.sh
   1. sudo service 6lbr start
   2. chromium
   3. save and exit
3. cd /lib/systemd/system
4. sudo nano 6lbr\_start.service
   1. [Unit]
   2. Description=Ensure 6lbr starts on boot
   3. After=multi-user.target
   4. [Service]
   5. ExecStart=sh /home/pi/6lbr\_start.sh
   6. User=pi
   7. [Install]
   8. WantedBy=multi-user.target
   9. Save + exit
5. sudo systemctl daemon-reload
6. sudo systemctl enable 6lbr\_start.service

Now onwards just power on the Raspberry Pi and the 6lbr will start running

Reference Links:

* <https://sunmaysky.blogspot.com/2015/08/setup-6lbr-to-run-6lowpan-with-cc2531.html>
* <https://sunmaysky.blogspot.com/2018/04/build-and-run-6lbrcontiki-ng-cc26xx-web.html>
* <https://github.com/cetic/6lbr/wiki/6LBR-Interface-Configuration>