
Hi!

This is an IEEE 802.11 a/g/p transceiver for GNU Radio that is fitted for operation with Ettus N210s and B210s. Interoperability was tested with many off-the-shelf WiFi cards and IEEE 802.11p prototypes. The code can also be used in simulations.

Table of Contents

1. Development
2. Installation
3. Usage
4. Troubleshooting
5. Further information

Development

Like GNU Radio, this module uses *maint* branches for development. These branches are supposed to be used with the corresponding GNU Radio branches. This means: the *maint-3.7* branch is compatible with GNU Radio 3.7, *maint-3.8* is compatible with GNU Radio 3.8, etc.

Installation

Dependencies

GNU Radio

There are several ways to install GNU Radio. You can use

- pybombs
- pre-compiled binaries
- from source

gr-foo

I have some non project specific GNU Radio blocks in my gr-foo repo that are needed. For example the Wireshark connector. You can find these blocks at <https://github.com/bastibl/gr-foo>. They are installed with the typical command sequence:

```
1 git clone https://github.com/bastibl/gr-foo
2 cd gr-foo
3 mkdir build
4 cd build
5 cmake ..
6 make
7 sudo make install
8 sudo ldconfig
```

Installation of gr-ieee802-11

To actually install the blocks do

```
1 git clone https://github.com/bastibl/gr-ieee802-11
2 cd gr-ieee802-11
3 mkdir build
4 cd build
5 cmake ..
6 make
7 sudo make install
8 sudo ldconfig
```

Adjust Maximum Shared Memory

Since the transmitter is using the Tagged Stream blocks it has to store a complete frame in the buffer before processing it. The default maximum shared memory might not be enough on most Linux systems. It can be increased with

```
1 sudo sysctl -w kernel.shmmax=2147483648
```

OFDM PHY

The physical layer is encapsulated in a hierarchical block to allow for a clearer transceiver structure in GNU Radio Companion. This hierarchical block is not included in the installation process. You have to open `/examples/wifi_phy_hier.grc` with GNU Radio Companion and build it. This will install the block in `~/.grc_gnuradio/`.

Check message port connections

Sometime the connections between the message ports (the gray ones in GNU Radio Companion) break. Therefore, please open the flow graphs and assert that everything is connected. It should be pretty obvious how the blocks are supposed to be wired. Actually this should not happen anymore, so if your ports are still unconnected please drop me a mail.

Python OpenGL

If you want to run the receive demo (the one that plots the subcarrier constellations), please assert that you have python-opengl installed. The nongl version of the plot does not work for me.

Run volk_profile

volk_profile is part of GNU Radio. It benchmarks different SIMD implementations on your PC and creates a configuration file that stores the fastest version of every function. This can speed up the computation considerably and is required in order to deal with the high rate of incoming samples.

Calibrate your daughterboard

If you have a WBX, SBX, or CBX daughterboard you should calibrate it in order to minimize IQ imbalance and TX DC offsets. See the application notes.

Checking your installation

As a first step I recommend to test the [wifi_loopback.grc](#) flow graph. This flow graph does not need any hardware and allows you to ensure that the software part is installed correctly. So open the flow graph and run it. If everything works as intended you should see some decoded packets on the console.

Troubleshooting

If GRC complains that it can't find some blocks (other than performance counters and hierarchical blocks) like

```
1 >>> Error: Block key "ieee802_11_ofdm_mac" not found in Platform - grc(
    GNU Radio Companion)
2 >>> Error: Block key "foo_packet_pad" not found in Platform - grc(GNU
    Radio Companion)
```

Most likely you used a different `CMAKE_INSTALL_PREFIX` for the module than for GNU Radio. Therefore, the blocks of the module ended up in a different directory and GRC can't find them. You have to tell GRC where these blocks are by creating/adding to your `~/.gnuradio/config.conf` something like

```
1 [grc]
2 global_blocks_path = /opt/local/share/gnuradio/grc/blocks
3 local_blocks_path = /Users/basti/usr/share/gnuradio/grc/blocks
```

But with the directories that match your installation.

Usage

Simulation

The loopback flow graph should give you an idea of how simulations can be conducted. To ease use, most blocks have debugging and logging capabilities that can generate traces of the simulation. You can read about the logging feature and how to use it on the GNU Radio Wiki.

Unidirectional communication

As first over the air test I recommend to try `wifi_rx.grc` and `wifi_tx.grc`. Just open the flow graphs in GNU Radio companion and execute them. If it does not work out of the box, try to play around with the gain. If everything works as intended you should see similar output as in the `wifi_loopback.grc` example.

Ad Hoc Network with WiFi card

- The transceiver is currently connected to a TAP device, i.e. is a virtual Ethernet interface. Therefore, we have no WiFi signaling like association requests and hence, the transceiver can not “join” an ad hoc network. You have to make some small changes to the kernel in order to convince you WiFi card to send to this hosts nevertheless.

-
- The transceiver can not respond to ACKs in time. This is kind of an architectural limitation of USRP + GNU Radio since Ethernet and computations on a normal CPU introduce some latency. You can set the number of ACK retries to zero and handle retransmits on higher layers (-> TCP).
 - RTS/CTS is not working for the same reason. You can however just disable this mechanism.
 - Currently, there is no CSMA/CA mechanism, but this can be implemented on the FPGA.

Troubleshooting

- Please check compile and installation logs. They might contain interesting information.
- Did you calibrate your daughterboard?
- Did you run `volk_profile`?
- Did you try different gain settings?
- Did you close the case of the devices?
- Did you try real-time priority?
- Did you compile GNU Radio and `gr-ieee802-11` in release mode?
- If you see warnings that `blocks_ctrlport_monitor_performance` is missing that means that you installed GNU Radio without control port or performance counters. These blocks allow you to monitor the performance of the transceiver while it is running, but are not required. You can just delete them from the flow graph.
- The message

You must now use `ifconfig` to set its IP address. E.g., `$ sudo ifconfig tap0 192.168.200.1`

is normal and is output by the TUN/Tap Block during startup. The configuration of the TUN/TAP interface is handled by the scripts in the `apps` folder. - Did you try to tune the RF frequency out of the band of interest (i.e. used the LO offset menu of the flow graphs)? - If 'D's appear, it might be related to your Ethernet card. Assert that you made the `sysconf` changes recommended by Ettus. Did you try to connect you PC directly to the USRP without a switch in between?

Further information

For further information please checkout our project page <https://www.wime-project.net>