

Massive MTC project: Guide

The project structure and compilation

Software files:

- **massiveUE**: Software with project software (this is where all the important stuff is)
- **srsUE**: The legacy version of srsUE that we branched from, when starting the massiveUE project. Only minor changes have been made (mostly debug messages for comparison with our own code).
- **srsLTE**: srsLTE library which is the major dependency library of srsUE/massiveUE. We have made changes to perform new functionality required in the implementation of massiveUE, e.g. some new buffer generation and more.
- **srsGUI**: Dependency package, I don't believe we have made any changes at all, simply just cloned and compiled back when we branched out of srsLTE's development path.

To compile the dependency files, you have to do the following:

```
$ cd srsLTE
$ mkdir build
$ cd build
$ cmake ../
$ make
$ sudo make install
```

```
$ cd srsGUI
$ mkdir build
$ cd build
$ cmake ../
$ make
```

You will have to use the *sudo* command for the *cmake/make/make install* commands.

Also, dependencies are required for these packages to install, see www.github.com/srsLTE/srsLTE (Build instructions). You will of course also have to install the [UHD driver](#) and you should make sure that you have [Volk](#).

With the project files, you will have to do the equivalent:

```
$ cd massiveUE (or srsUE)
$ mkdir build
$ cd build
$ cmake ../
$ make
```

Please note that they have made major changes in their structure since we branched out, namely their "srsUE 2.0" release which was released at the day we handed in our thesis. You will have to investigate this yourself therefore – one major change is that srsLTE and srsUE libraries have merged.

It seems like they have updated their install-guide etc. in their GitHub readme (now collectively within the srsLTE project and not split between srsLTE/srsUE); especially the installation guide was somewhat outdated when we started.

To run our code, you must go to `./build/ue/src/` and run:

```
$ sudo ./run_ue ue.conf
```

In the legacy srsUE, you should go to the same folder and run:

```
$ sudo ./ue ue.conf
```

Short introduction to the project files

Within our project, the files are within the `./ue` folder in the header (*hdr*) folder and source code folder (*src*). Our implementation is within the radio layer (*radio*) and physical layer (*phy*) modules, as well as the `ue` module (*ue.cc*) module and main module. The “device handler” is called *ue_handler* in the code and is defined in its own module (*ue_handler.cc*). Some changes have also been made in the gateway module (*upper/gw.cc*) to enable the generation of a TUN device for each instantiated device.

The amount of devices is set in *main.cc* on the first line of the main function (l. 369) before compilation. The “common physical layer” (coPHY) as we devised it in our project is developed out of the existing “*phch_rcv*” (“physical channel receiver”) module. The module has kept its name in the source code, i.e. *phch_rcv*, and it is instantiated under the name *sf_receiver* (subframe receiver).

Short list of the major structural changes we have made in the library modules:

- Radio layer is instantiated and is set up before devices and not within the set-up of the/a device.
- Subframe receiver/phch. receiver/coPHY is also instantiated on its own before individual devices are instantiated and set-up.
- The device handler (*ue_handler*).

You will also see, that some function calls with go through e.g. the *phy.cc* module from other phy-layer modules. This is only due to the fact that this module “knows” the device id (*ue_num*). This is so that the device id is not to be registered in all modules, requiring a lot of changes in all modules, but also (mainly) to maintain a relatively low amount of modules trying to use the mutexes in the radio layer. Therefore, we have changed some calls that before went from “upper layers” (above radio) directly to the radio layer, so that they now take a detour and can be handled by the general physical layer module.