5G-AND-BEYOND Prototyping

FINAL REPORT

Team: SDMAY22-30
Client/Advisor: Hongwei Zhang
Members:
Josh Guyer / Hardware Lead
Josh Naber / Hardware Team
Raffael Neuser / Hardware Team
Johnathan Leisinger / Software Lead
Connor Kesterson / Software Team
Ruofeng Gao / Software Team
Nicholas Garrelts / Software Team
sdmay22-30@iastate.edu
Website: sdmay22-30.sd.ece.iastate.edu

Revised: 4-24-22/V.1

Table of Contents

1. Introduction

1.1 Problem Statement

The problem that we were tasked with is that the ARA Wireless network is in need of a portable enclosure which can be used to deploy the necessary components needed to attach to the ARA network in rural areas. ARA stands for Agriculture and Rural Communities. The ARA Wireless Network is an at-scale platform that will be utilized for advanced wireless research across rural communities. Some examples of the places that will need such enclosures deployed include farms as well as CyRide buses.

1.2 Solution

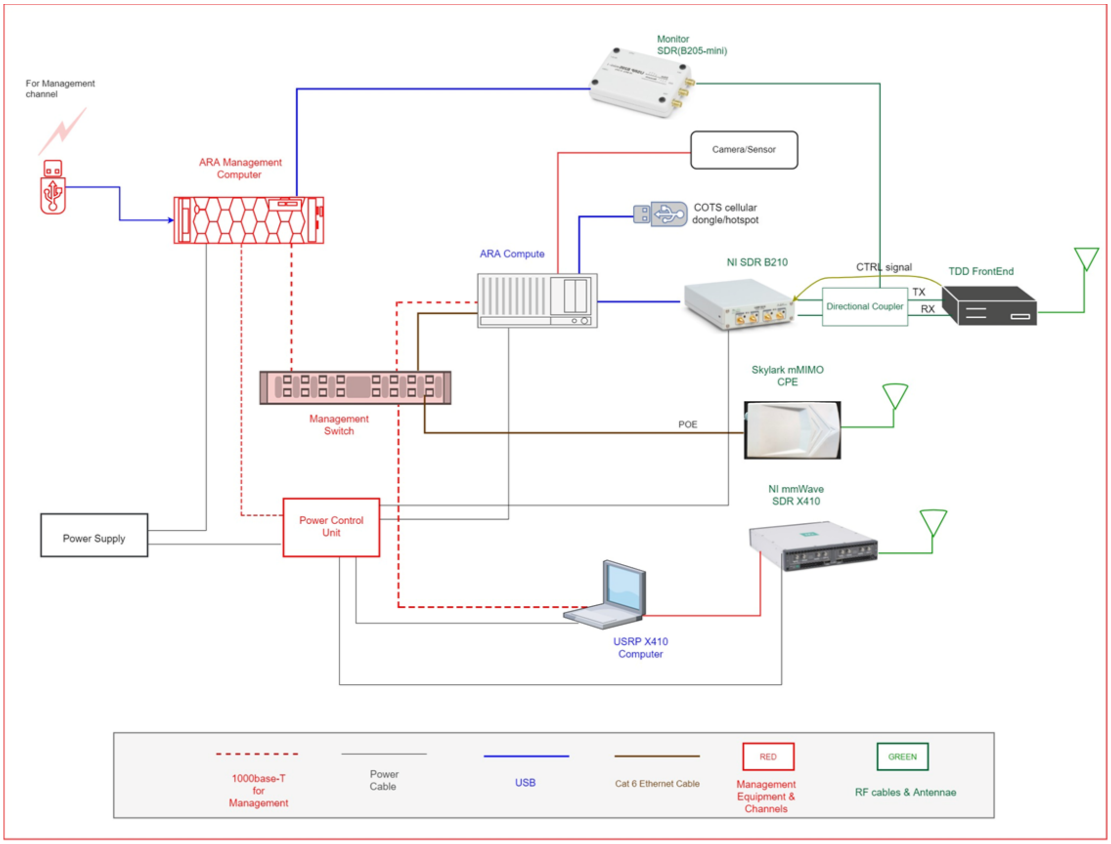
Create an enclosure that will house all the necessary components for 5G user equipment to communicate with a base station. Also modify open source software to increase performance metrics of the software defined radios (SDR).

1.3 Intended Users

Users: Universities, farmers, rural communities, and the city of Ames
Uses: Provide connectivity for wireless networks, cloud computing, plant phenotyping, remote monitoring, and tele-operation of agricultural vehicles

1.4 System Design

Pictured below is the block diagram for how the enclosure is set up. Inside the enclosure is a management computer, compute computer, B205 SDR, B210 SDR, Skylark SDR amplifier, network switch and a power supply. These components are connected following the diagram. The management computer is configured to be a DHCP server so the entire UE has its own LAN. Since the computers are inside an enclosure, they are accessed using SSH through the management computer using the B205 SDR. This allows for remote access, even if there is no WiFi available. In the future, they plan on adding a cellular router to the enclosure which would provide the same service.



2. Evolution of design

2.1 Enclosure design changes

We began our design process last semester with the idea of 3D printing the enclosure because we believed that it would allow for fast prototyping and was low cost. This changed as we were told that more components would go inside the enclosure and 3D printing was limited in size. We changed our design to use a premade waterproof container and focused on planning how the components inside would be laid out. We continued the 3D modeling strategy of designing the enclosure and worked with a manufacturer at Iowa State who was able to cut acrylic panels using a waterjet. The manufacturer assisted us with the design process and provided valuable insight into how the enclosure should be designed. In addition to the design change, many of the components inside the enclosure changed over the semester as the graduate team we were working with found different devices. Everytime they changed the components, we had to modify how they would be laid out which made developing continuously difficult. Additional modifications are going to be made to the enclosure in the future by the graduate team.

2.2 Software design changes

This semester we identified specific algorithms to implement into srsRAN: UCS, LDP, D2D, and WFQ. We provided comparative analyses of the different algorithms and connected the enclosure in a testbench environment. The software testbed environment was taken offline since last semester, so we decided to use the enclosure prototype in our testing. Our design has shifted from testing the prototype in a field environment to an indoor lab test due to hardware time delays.

3. Requirements

3.1 Functional requirements

- Deploy Agriculture and Rural (ARA) base station equipment
- Perform measurements of network reliability and speeds

3.2 Non-functional requirements

- Enclosure should be portable
- Enclosure should be easy to manufacture
- Prototype should be easy to deploy
- The algorithm should improve the performance of the SDRs
- Minimize the average latency
- The algorithm should improve the transmission efficiency

3.3 Technical requirements

- Experiment with 5G-and-beyond solutions
- Implement srsRAN source code working
- The stable transmission
- 3D printing materials to print enclosure
- Combine software and hardware

4. Industry Standards

Industry standards we used on this project are IEEE 802 and IEEE 1914.1. IEEE 802 is a networking standard used so that we can define certain benchmarks for our system in terms of network latency and reliability standards. IEEE 1914.1 is a radio protocol standard that is used to describe the architecture of the network system like how our testing environment has a base station communicating to user equipment.

5. Engineering Constraints

The constraints of this project are based on the functional and nonfunctional requirements. Since the enclosure needed to be water resistant and portable, we had to find something that fit those requirements and was also not expensive. The enclosure needed to be easy to work in and something that we could buy in bulk as they are going to need more of them. We were also constrained by time. A product like this requires years of iteration and development to get a clean final product. With more time, we would have been able to iterate the enclosure more, improving on some of the inconveniences of it.

6. Security concerns

The enclosure itself needs to be physically protected from break ins because the components inside are expensive. The latches on the bottom of the enclosure have holes in them which allow for a padlock to be put on them. This prevents physical entry to the enclosure. Remote entry and tampering is also a problem because research groups will be using this system for wireless experiments. To prevent remote tampering, a management computer is in the enclosure which only the ARA research group has access to. This computer will be able to change things on the research computer if someone got access and broke something.

7. Implementation

7.1 Hardware implementation

The design of the enclosure consists of 3 levels made of quarter thick acrylic. These panels are cut using a waterjet and assembled with the components using screws and nuts. The bottom panel has the Dell Precision 3240 computer and a power strip which is soldered to the power entry module on the enclosure. The second panel has the Intel NUC, B205 SDR, and netgear 8 port switch. The switch connects to both computers and all the ethernet ports on the outside of the case. The top panel has the power supplies for the computers, the B210, the amplifier, and the transformer. There were holes cut in the panels to allow for wiring between levels. Refer to the operation manual in section 8.1 on how to manufacture an enclosure.

7.2 Software implementation

The software design consists of the hardware team's enclosure with a computer and B210 SDR, which acts as the user equipment (UE) and an external computer and X310 SDR that is the base station running the evolved packet core (EPC) and eNodeB (ENB). The ENB is directly communicating with the UEs and the EPC aggregates the data from the ENB. The SDRs are configured to have the same uplink and downlink frequencies so they are communicating in the same frequency range. Then, modified source code can be loaded onto the base station and UE computers to test changes in latency and reliability compared to baseline srsRAN code.

Implementing algorithms into srsRAN:

The UCS (Unified Cellular Scheduling) algorithm was found in the paper "Unified Scheduling for Predictable Communication Reliability in Cellular Networks with D2D Links" by Yuwei Xie, Hongwei Zhang, and Pengfei Ren. In order to implement the UCS algorithm into srsRAN there are a few changes that need to be made from LTE standards. A few of these changes come in the PDCCH (physical downlink control channel) and PUCCH (physical uplink control channel). A local signal feedback flag as well as a local terminal indicator need to be added in the PDCCH. As for the PUCCH, a local signal map indicator is needed for when the UE receives the local signal map update from the base station. The other modification that needs to be made is expanding the X2 interface so that it can indicate precisely the links between UE's in different cells. This can be done by adding a new X2 message PRK-Signal.

Local-deadline partition (LDP) is a scheduling algorithm from the paper "Multi-Cell, Multi-Channel Scheduling with Probabilistic per-Packet Real-Time Guarantee" by Meng, Zhibo, and Hongwei Zhang. LDP has local deadlines and local work densities to determine priority and provide per-packet reliability. It shares a similar implementation to UCS with the following additions: a local packet deadline and a calculated local work density. These additions are shared with other links to determine the priority levels to and the order in which packets should be transmitted.

The Device-to-Device (D2D) algorithm was found from the paper titled, "Algorithm and Scheme for D2D Communication in 4G/5G Networks" by Rosalee et al. The D2D algorithm allows a transmitting UE and receiving UE to communicate directly with each other without going through the base station. This is done by allowing this pair of UEs to only communicate on the frequencies that are both shared between the UEs and have low interference due to other devices operating in the same frequency band. The key variables used in this algorithm are the number of UE pairs, number of interfering devices, distance in meters of each UE and interfering device, carrier frequency in kHz, in-band, or licensed, bandwidth, out-of-band, or unlicensed, bandwidth, transmit gain of the base station, transmit gain of the paired devices, and transmit gain of the interfering devices.

The Weighted fair queueing(WFQ) is a network scheduling algorithm from paper "Performance of weighted fair queuing systems with long range dependent traffic inputs " by M. Ashour and Tho Le-Ngoc. Weighted fair queueing is also known as packet-by-packet GPS. Base on the introduction of WFQ, i try to use it in srsRAN and make the transmission more efficient and stable. WFQ is a dynamic process that divides bandwidth among queues based on weights. The process is designed to be fair, such that WFQ ensures that all traffic is treated fairly with regard to its weight.

8. Testing process and testing results

For the hardware side of our project, the testing really came down to trying to physically put together our system and see how it all fit together. One of the aspects that we had to test a lot was how the acrylic panels fit inside our enclosure. This form of testing required us to get the cut acrylic panels and test to see if they fit snugly in the enclosure. This meant that we were able to put the panels in the enclosure, have the corner holes lined up to screw it into the enclosure, as well as have a way to get the panel out of the enclosure (such as handles on the side). Another thing we had to test was how to attach all of the components we had onto the acrylic panels. This process included us getting the screw holes cut in specific positions on the panel as well as testing different standoffs and screws to securely attach the components. The last thing we had to test on the hardware side was the connections between all of the components. This required us to test different ethernet cable lengths as well as different SMA cables to ensure that everything was connected without a ton of excess cabling.

As a result of these testing processes, we were able to create acrylic panels that snugly fit in our enclosure that are easy to handle and extract from the enclosure. We were also able to ensure that our components attach tightly to our panels as well as are all connected by efficient cabling.

For testing the software, we set up an X310 to work as the base station along with the hardware team's enclosure containing a B210 to act as the UE. Pictured below is a screenshot of a successful connection of the BS to the UE. The BS is receiving data being transmitted by the UE, so the configuration files were correctly altered to allow the UE to attach to the BS searching for connections. We haven't been able to test any algorithm implementations on the physical testbench setup as we were unable to get the testbench fully operational. A local testing environment is supported by srsRAN, however. With the use of ZeroMQ, radios and networks can be configured on one computer to simulate multiple networks and devices. ZeroMQ also provides integration support for GNU-Radio which can be used to introduce arbitrary signals and noise between radios, or for signal processing.

For an application a peripheral sensor, like a camera can be connected to the enclosure which can be out in the field and the transmission is sent to a centralized BS and a user can see the video feed remotely or other transmissions from multiple UEs.

```
==== eNodeB started ===
Type <t> to view trace
Setting frequency: DL=2620.0 Mhz, UL=2500.0 MHz for cc_idx=0 nof_prb=50
RACH:  tti=9141, cc=0, preamble=19, offset=1, temp_crnti=0x46
User 0x46 connected
t
Enter t to stop trace.

          ----------------DL---------------|-----------------------UL-----------------------
rat rnti  cqi  ri  mcs  brate   ok  nok  (%) | pusch  pucch  phr  mcs  brate   ok  nok  (%)    bsr
lte   46   10   0    0      0    0    0   0% |   n/a    1.6    0    0      0    0    0   0%    0.0
lte   46   10   0    0      0    0    0   0% |   n/a    2.6    0    0      0    0    0   0%    0.0
lte   46   10   0    0      0    0    0   0% |   n/a    1.7    0    0      0    0    0   0%    0.0
lte   46   10   0    0      0    0    0   0% |   n/a    2.4    0    0      0    0    0   0%    0.0
lte   46   10   0    0      0    0    0   0% |   n/a    2.8    0    0      0    0    0   0%    0.0
lte   46   10   0    0      0    0    0   0% |   n/a    2.7    0    0      0    0    0   0%    0.0
lte   46   10   0    0      0    0    0   0% |   n/a    2.3    0    0      0    0    0   0%    0.0
lte   46   10   0    0      0    0    0   0% |   n/a    2.9    0    0      0    0    0   0%    0.0
lte   46   10   0    0      0    0    0   0% |   n/a    2.9    0    0      0    0    0   0%    0.0
lte   46   10   0    0      0    0    0   0% |   n/a    1.9    0    0      0    0    0   0%    0.0
lte   46   10   0    0      0    0    0   0% |   n/a    0.9    0    0      0    0    0   0%    0.0
```

9. Related literature

"Ara Wireless Living Lab." *ARA*, https://arawireless.org/.

"5G Mobile Networks: A Systems Approach¶." *5G Mobile Networks: A Systems Approach - 5G Mobile Networks: A Systems Approach Version 1.1-Dev Documentation*, https://5g.systemsapproach.org/.

Xie, Yuwei, Hongwei Zhang, and Pengfei Ren. "Unified scheduling for predictable communication reliability in cellular networks with D2D Links." *Computer Communications* 167 (2021): 1-14.

Meng, Zhibo, and Hongwei Zhang. "Multi-Cell, Multi-Channel Scheduling with Probabilistic per-Packet Real-Time Guarantee." *ArXiv.org*, 19 Mar. 2022, https://arxiv.org/abs/2101.01768.

Zhang, Hongwei, et al. "Scheduling with predictable link reliability for wireless networked control." *IEEE Transactions on Wireless Communications* 16.9 (2017): 6135-6150.

Chen, Yu, et al. "Probabilistic per-packet real-time guarantees for wireless networked sensing and control." *IEEE Transactions on Industrial Informatics* 14.5 (2018): 2133-2145.

Bennett, Jon CR, and Hui Zhang. "WF/sup 2/Q: worst-case fair weighted fair queueing." *Proceedings of IEEE INFOCOM'96. Conference on Computer Communications*. Vol. 1. IEEE, 1996.

10. Appendices

10.1 Operation manual

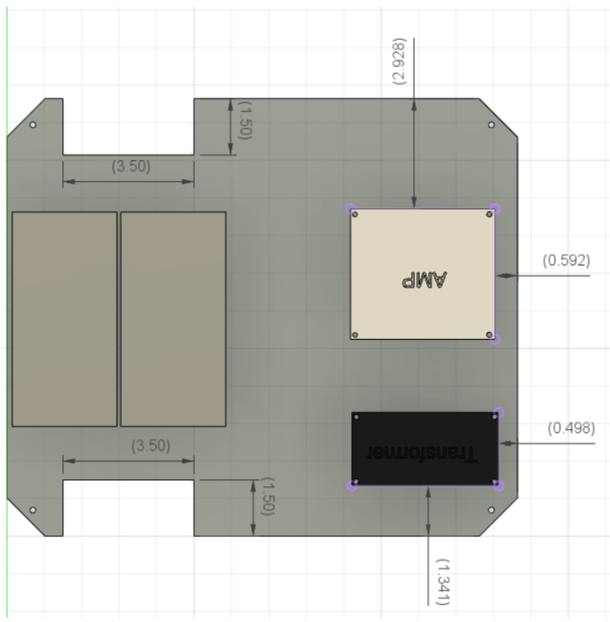
As we finished our project, we created an operation manual for how to create another enclosure because future senior design groups or teams will need to make more.

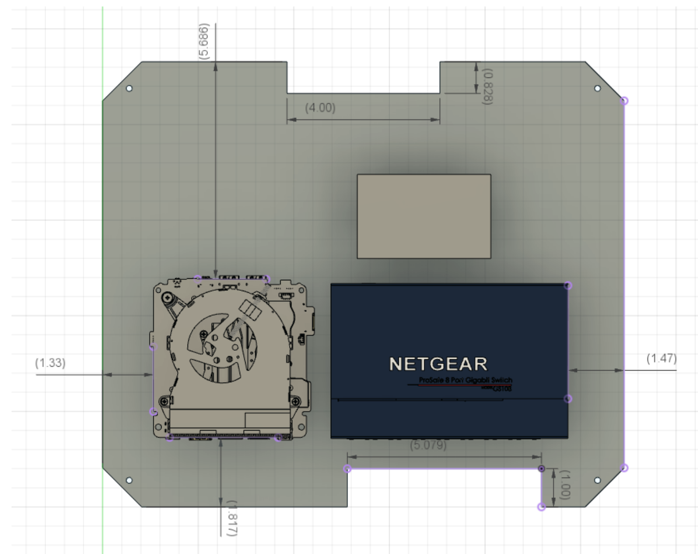
How to create/setup the enclosure:

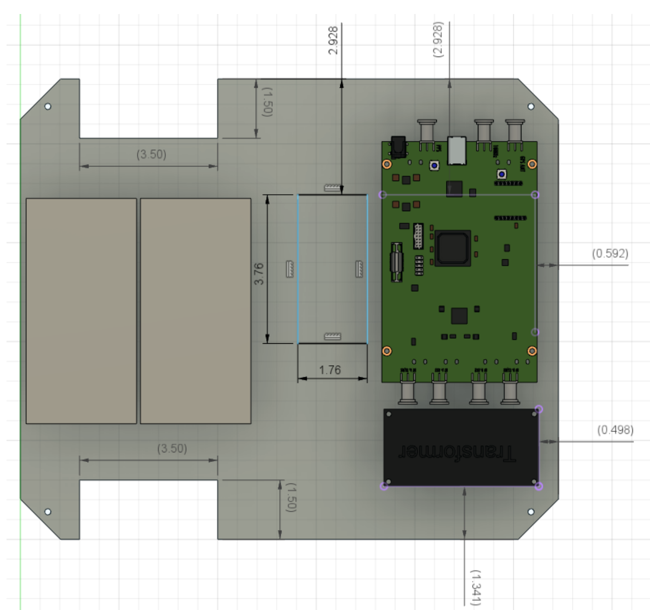
1. Get the panels cut from Hoa in the ABE lab. From our experience, he may have excess acrylic he is willing to use. If not, you will have to provide the acrylic yourself. (Source files for the acrylic panels can be found on our senior design website)
2. Put all the components on the panels. The bottom panel should have the Dell 3240 and the power strip, the middle panel should have the switch, and NUC, and the top panel should have the power supplies, B210, amplifier, and transformer.
    a. NUC
        i. The NUC goes on the middle panel in the bottom left corner. It uses 25mm screws and 10mm standoffs. Put the screw through the bottom of the panel and put a 10mm standoff on the other end. Orient the NUC such that the ethernet port is facing the back of the enclosure. Put the NUC on the screws and then put 4 nuts on top of the screws. Don't screw the nuts too tight or they will bend the NUC.
    b. Switch
        i. The switch goes next to the NUC on the middle panel. The switch uses 2 20mm screws and 4 nuts. First put the screws on the switch by putting a screw into the mounting hole and tightening a screw onto it. Orient the switch such that the ethernet ports face the outside of the enclosure. Then put the switch on the panel and tighten another nut on the underside of the panel.
    c. Transformer
        i. The transformer is very similar to the NUC in the mounting method. It goes on the top panel. Depending on the transformer used, it either goes in the bottom right corner or in the middle. It should use 20mm screws and 5mm standoffs.
    d. Amplifier
        i. The amplifier goes below the B210 on the top panel. It uses 4 25mm screws on the corners. Put the screws through the bottom of the panel and through the amplifier, then put nuts on top. Orient the amplifier such that the SMA ports are facing the inside of the enclosure.
    e. B210
        i. This goes on top of the amplifier using 30mm standoffs. Use 20mm screws to put the standoffs on the top of the panel. Then put the B210 on top and fasten it with 5mm screws. Orient the B210 so that the USB port is facing the front of the case.
    f. Dell Precision 3240
        i. The Dell goes on the bottom panel and uses M4 screws. It should be oriented such that the back of the computer faces the power strip.
    g. Power strip

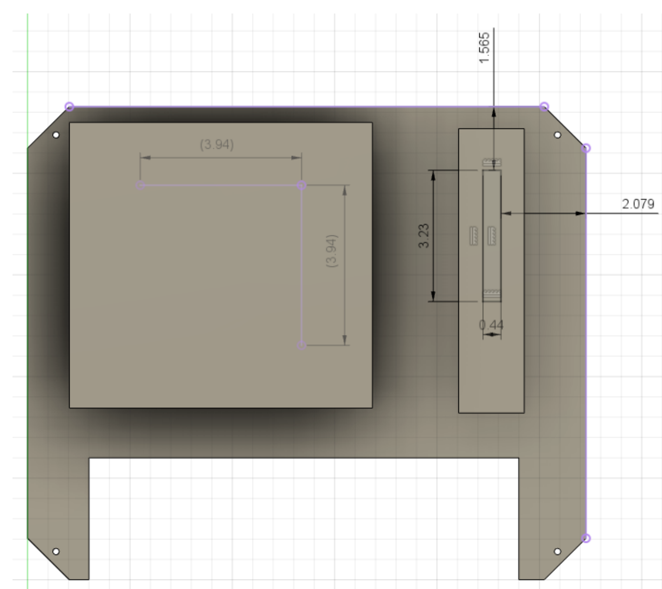
        i. Mounted in a similar way to the switch. First put the head of the screw in the holes of the strip then tighten a nut on the screw. Put the screws through the panel and tighten another nut on.

3. Cut the holes in the enclosure for vents, ethernet ports, and antennas
    a. Cut 4-5 holes in the front of the enclosure on the bottom for ethernet ports. These holes should be M20 in size, big enough to fit the threading through but not bigger than the gasket. A hole should also be placed on the bottom for the power entry module. Two holes for the vents should be on the sides on the top end. It should be two square holes for the vents and 4 holes in the corners for the screws. Finally, two holes need to be cut in the top of the case for the SMA cables used for the antennas.
4. Splice the ethernet cables
    a. You will need to make your own ethernet cables to length for the connection between the NUC and the switch, Dell and the switch, and POE and the switch. Here is a walkthrough video on how to properly splice ethernet cables.
5. Solder the power connections
    a. Currently the model only uses a power strip without a relay. The power strip cable needs to be cut and soldered to the power entry module. Use the detachable wire connectors to connect the power entry module to the power strip. Green is ground, black is load, and red is neutral. It is VERY IMPORTANT that you match the wires from the power entry module to the strip. Also use heat shrinks to protect the soldered wire connections.
6. Put the hinges in the corners of the enclosure
    a. The hinges go in the corners of the enclosure. The bottom panel screws directly into the bottom and doesn't need hinges.
7. Connect all the components
    a. This is the most complicated part of the assembly. All the components need to be connected in a specific order or else you won't be able to put the next panel on. On the bottom panel, make sure all power cables are connected to the power strip, the ethernet cable and USB cable for B210 are connected to the Dell, the power strip is connected to the power module, and the Dell is connected to its power supply. Then put the second panel on, keeping the power cables on the side. Connect all the ethernet ports to the switch as well as the cables from the Dell and NUC, and connect power to the switch. Put the final panel on and connect the power to the transformer, USB to the B210, power cables to the power supplies for the NUC and Dell, and antennas from the AMP to the side of the panel.


(2.928)
(1.50)
(3.50)
(0.592)
(3.50)
(0.498)
(1.50)
(1.341)


(5.686)
(0.828)
(4.00)
(1.33)
NETGEAR
(1.47)
(5.079)
(1.00)
(.817)


2.928
(2.928)
(1.50)
(3.50)
3.76
1.76
(0.592)
(0.498)
(3.50)
(1.50)
(1.341)


1.565
(3.94)
(3.94)
3.23
2.079
0.44

Setting up srsRAN:

The code below is from Installation Guide — srsRAN 21.10 documentation. Further details can be found by following the hyperlink.

sudo add-apt-repository ppa:softwareradiosystems/srsran

sudo apt-get update

sudo apt-get install srsran -y

sudo apt-get install build-essential cmake libfftw3-dev libmbedtls-dev libboost-program-options-dev libconfig++-dev libsctp-dev

(install RF front end driver — in our case, UHD)

git clone https://github.com/srsRAN/srsRAN.git

cd srsRAN

mkdir build

cd build

cmake ../

make

make test

sudo make install

srsran_install_configs.sh user

Refer to Section 10.4 for configuration parameters that need to be changed to create a valid connection between BS and UE.

How to run the code:

to connect to base station and ue: ssh root@[IP address]

base station IP: 10.24.100.14
Following commands should be ran in separate terminal windows:

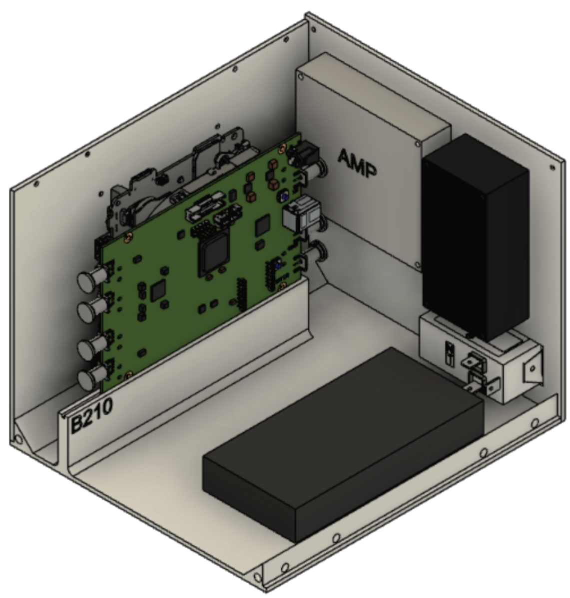
1. srsepc - defines terminal window as intermediary between base station and ue
2. srsenb - defines terminal window as base station

UE IP: 10.48.163.240

1. srsue - defines terminal window as ue

10.2 Alternative versions of design

The first design iteration that our project went through involved 3D printing an enclosure to house all of the necessary components. After going through many iterations, we decided this wasn't the best way to go about solving this problem due to the constantly changing components within the enclosure. We also found out quickly that 3D printing really limits the size of an enclosure you can make, so that really deterred us from this path. The research group is also going to need many of these enclosures manufactured and 3D printing would not work on a large scale due to the long printing times for some of the parts.



Another design iteration that we went through was deciding the size of our enclosure. After we moved from 3D printing to buying a pre-made waterproof enclosure, we still ran into a problem of changing components within the enclosure. This required us to order a bigger enclosure and design the inside of it a little differently due to these changes.

10.3 Other Considerations

Designing the enclosure was a difficult task for us because none of us have experience with 3D modeling or manufacturing. This meant that we spent a lot of time just figuring out the next steps to take to create this enclosure. We would always find something that we forgot to consider when designing. If we were to go back and repeat this process, we should talk with mechanical engineers with relevant experience to get their input for the design.

10.4 Code

To change configuration files for linking BS and UE:

cd .config/srsran

For UE: nano ue.conf

In RF section: Set tx_gain to 80 and make sure rx_gain is commented out

```
[rf]
freq_offset = 0
tx_gain = 80
#rx_gain = 40
```

In rat.eutra section: set dl_earfcn = 2750, ul_freq = 2500e6, dl_freq = 2620e6

```
[rat.eutra]
dl_earfcn = 2750
ul_freq = 2500e6
dl_freq = 2620e6
#nof_carriers = 1
```

For BS: nano enb.conf

In RF section: Set tx_gain to 80, make sure rx_gain is commented out, set dl_earfcn = 2750, ul_freq = 2500e6, dl_freq = 2620e6

```
[rf]
dl_earfcn = 2750
tx_gain = 80
#rx_gain = 40
ul_freq = 2500e6
dl_freq = 2620e6
```