Lab 3

Deadline: 11:59 PM on Friday, Feb 28, 2020

Hardware / Sensors

- 2 x GNSS/ RTK Processing boards
- 2 x GNSS Antennas
- 2 x 915 MHz Telemetry Radios (except for teams with ublox c009-f9p boards)

Data collection Policy

Plan data collection schedules amongst your teammates. You can talk to your teammates and ask questions on Piazza.

- Data acquisition can be done collectively in a team. One dataset per team is sufficient.

- Analysis on the collected dataset, should be done **individually**.

Hardware Setup

The RTK GNSS system given to a team can be one of the three different products as listed below. Check the appropriate datasheet and appendix for the sensor you have received.

Product	Processing Board	Radio Connection	Appendix
EMLID reach	Ublox neo-m8t	915 MHz Radio pair	Appendix A, B
UBLOX C009-FPP	UBLOX ZED f9P	Use onboard wifi hotspot	Appendix C

After setting up , you should have a system where you have the base RTK, rover RTK and a laptop working together. The base is stationary and sending corrections to the rover over radio and the laptop is connected to the rover via usb-serial to receive the corrected gnss fix solution in NMEA string.

Appendixes List

A: Testing telemetry Radios.
B: Emlid reach rtk setup
C: Ublox c009 - f9p eval board

All the sensors have to be configured to output NMEA string as output on usb-serial, so parsing

it will be similar to Lab 1. Make sure, your ros msg will contain the status of the gnss Fix quality (from the nmea string 'GGA') to advise whether your fix is either gnss fix, rtk float mode, rtk fix. It will be useful for analysis.

What to Submit for Lab 3 ?

1. **Data collection**

   Go Outside, Setup your rtk pair, Average your base station(Don't move it after setup), Make sure your rover is in either RTK float or RTK fix mode.

   Collect a 10 minutes stationary data set and another dataset with the ROVER moving in a structured path (eg. forward X mts, right Y mts and back X mts etc.) coming back exactly to the same starting point with the same heading. Collect each of these data sets at the following locations:
   a. At a spot with partial occlusion and reflections nearby eg. Outside of ISEC
   b. At a spot with which is completely clear eg. open park or roof of Columbus parking garage

2. **Analysis**

   Examine the utm data (by plotting it or doing statistics on it)

   What does this say about RTK GNSS navigation? Look at the error estimates. What can you say about the distribution of noise in the signal? Repeat this for each of the datasets collected.

   Write this lab up into a report in pdf format, highlighting your analysis.

Grading Rubric (10 Points)

- 1 points for working device driver with fix status
- 3 points for data collection
- 4 Points for Analysis
- 2 Points for Report

Late submission policy is mentioned in the syllabus.

How to Submit Lab_3

1. In your class repo 'EECE5554_RoboticsSensing', create a directory called LAB3
2. Copy the ros driver package used for this assignment under LAB3.
3. Inside LAB3, create sub-directory 'analysis'
4. Copy the matlab/ python code used for data analysis under 'analysis'
5. place your report in pdf format also in analysis directory. (also upload the report in pdf format on blackboard).

An example repo structure is available at
https://gitlab.com/PushyamiKaveti/eece5554_robotsensing.git

Your repo structure should look similar to

'<Path_to_repo>/EECE5554_RoboticsSensing/LAB3/<your_gnss_ros_driver files>'
'<Path_to_repo>/EECE5554_RoboticsSensing/LAB3/analysis/<your analysis files>'
'<Path_to_repo>/EECE5554_RoboticsSensing/LAB3/analysis/report.pdf'

6. Push your local commits to (remote) gitlab server. You can verify this by visiting gitlab.com and making sure you can see the commit there.
7. Upload your report in pdf format to Blackboard Assignments under Lab_3

(Appendixes is on Next Page)

appendixes

Appendix A: Testing telemetry Radios

All the 915 MHz sik based telemetry radios are paired and will get connected by default. Radio pairs with the same number label(eg: 22 label on both radios) are paired.

LED 1 :
A Green LED on the radio indicates paired. Blinking green LED indicates searching for a paired device. A solid RED indicates failure, restart your device. Contact TA if needed.
LED 2:
A Blinking RED led indicates data transfer

Checking connection between two radios:

Connect the first radio with usb cable to your laptop, check its port name on $dmesg in your terminal. Should be something along '/dev/ttyUSB*'. Now open minicom with port and baud rate of 57600, 8N1

Now connect the second radio either to the same laptop or another laptop, check its port name and open minicom in a different terminal with 57600, 8N1 settings.

Now if you type in one minicom terminal attached to a radio, same characters should appear on the second minicom terminal connected to the second radio. As you type in terminal you can see LED 2 blinking red indicating data transmission.

If you see solid Green on led 1, both radios are connected.
If you dont see blinking Red on led 2 with solid green on led 1, then check baud rate settings on both terminals.

Appendix B: Emlid reach rtk setup

@ Instructions for setting up Emlid Reach RTK
@ By Yang Liu in 2019 (slightly modified)
useful link: https://docs.emlid.com/reach/
Download links for reachview app are below(optional)
Ios: https://itunes.apple.com/us/app/reachview/id1295196887?mt=8
Playstore: https://play.google.com/store/apps/details?id=com.reachview
#The app is an easy way to debug reach devices in the field, and can be done on a laptop as well.

Setting up the RTK Rover:
1. All reach devices are equivalent, so pick one as a rover and connect it to your laptop through a USB port. Now, connect the radio and reach through UART. A red light will turn on when your rover is still initializing. If the radio boots up correctly, a blinking green light will take the place, indicating that it's searching for its paired radio. A fully connected gps rover should look like this:

NOTE: Some models of laptops constrain the power output of their USB ports (e.g. XPS-15), so check if your USB port can power up both reach and radio properly. If the voltage is too low, the red light will not turn green on the radio. **DO NOT** power up the reach and the radio separately and then connect them, it will damage the devices! Contact TA if you are having the power issue.

2. When reach is powered up properly (may take some time), it goes into hotspot mode (if no known Wi-Fi is detected) and broadcasts a hotspot named as **reach:xx:xx**. The default password is **emlidreach** (please don't change default name and password).

3. Once you connect to the hotspot successfully, go to **http://192.168.42.1** in the browser. You will see a website showing the device's status.

4. On the device website, check if the software version is v2.16.2. If not, please contact TA. Here are some settings you want to check before saying hi to the cold wind in Boston. **Please refer to Appendix B.1 for details**.

5. The device can also connect to wifi networks, don't do it.
    #Please do not connect reach devices to nuwave or eduroam. Port forwarding is restricted on nuwave and so once device connects to nuwave the only options to get back to the reach webpage is to take the device off campus where there is no nuwave signal, forcing the device into hotspot mode.

Set up the RTK Base:

Now boot up the other reach as your rtk base. It's almost the same to rover gps, just that you don't have to use the USB port on your laptop as power source since you are not gathering any data from a rtk base. Similar to setting up a rtk rover, you need to connect to the hotspot of the device and check some configurations. **Details are in Appendix B.2**.

Tip: If you are confused by which device you are connecting to: click the light bulb on the main page (192.168.42.1) of the device, it will force the LED on the device to blink for a while, so that you know whose Wi-Fi you are connecting to.

## Check the Connection between rover and base:

The connection between rover and base have to be tested in open space. If the rover is receiving data from the base, the state page from the rover should look like this:
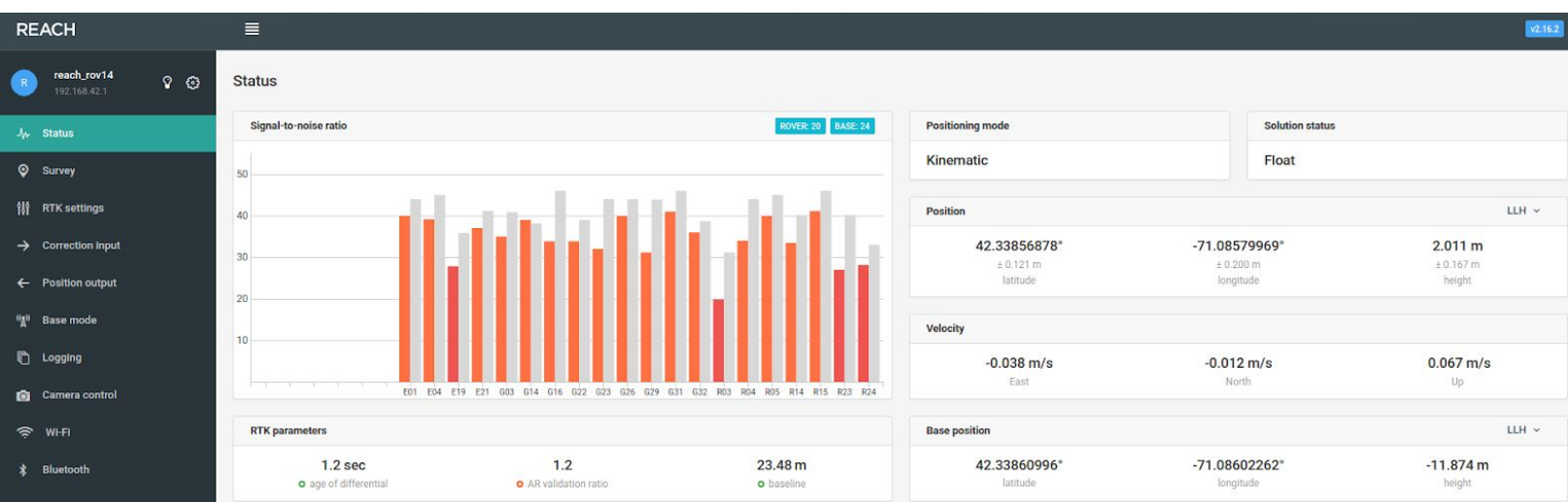

Notice that you can see the satellite information on the base side, too. The solution status turns usually from **single -> float -> fix**. Now open minicom in a terminal, you will be able to see the output similar to that in lab 1. When the base has finished accumulating its coordinate, the longitude and latitude values in GPGGA messages should be very stable. Usually, only the last two (occasionally three) digits change.
Congratulations, your rtk gps system is all set!
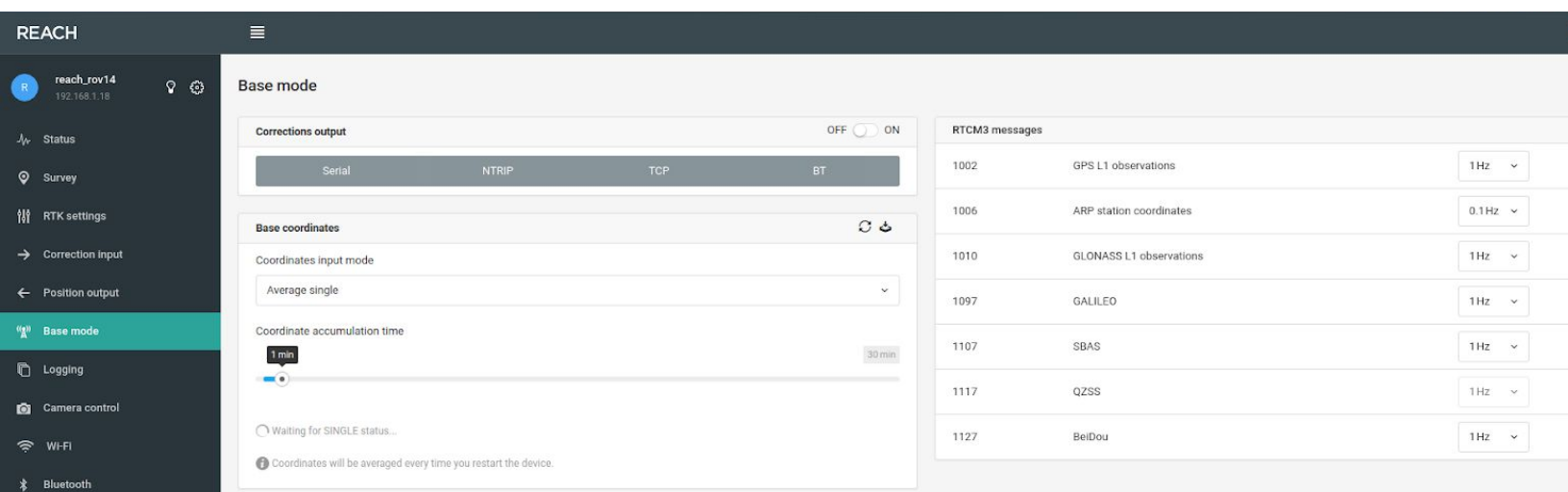
Appendix B.1
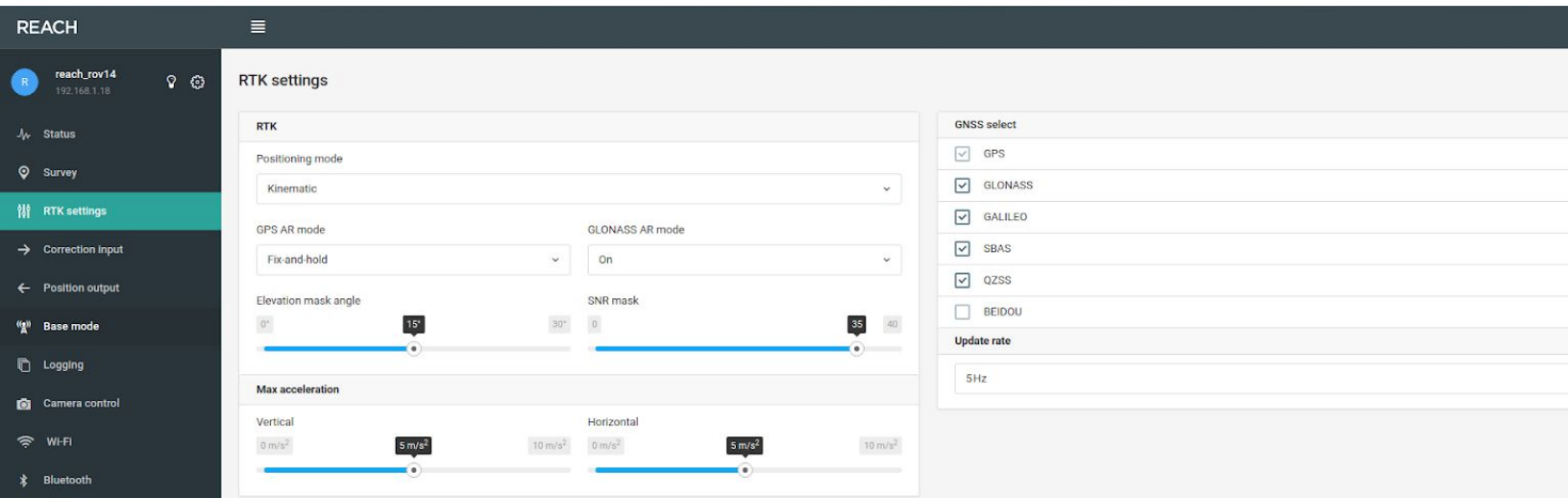
1. In **RTK Settings tab**, check as many GNSS systems as possible and set the update rate to 5 hz. Positioning mode should be set to Kinematic.
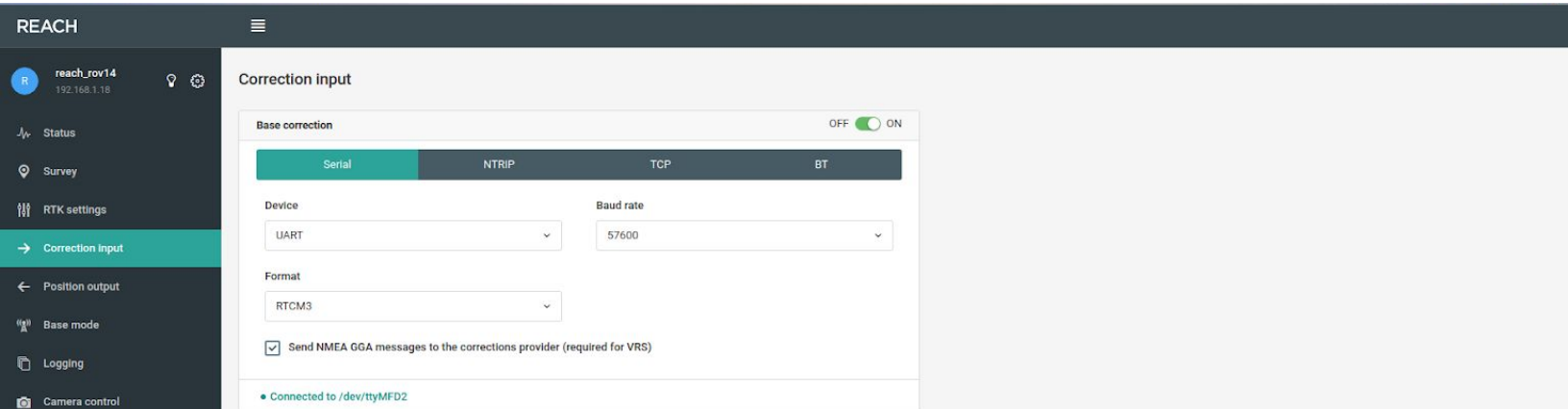
2. In **Correction Input tab**, you can configure how the rover get position error correction from the rtk base here. You are getting correction data through UART from radio, so please set it to Serial mode. All settings on this page should be the same as shown in the screenshot.
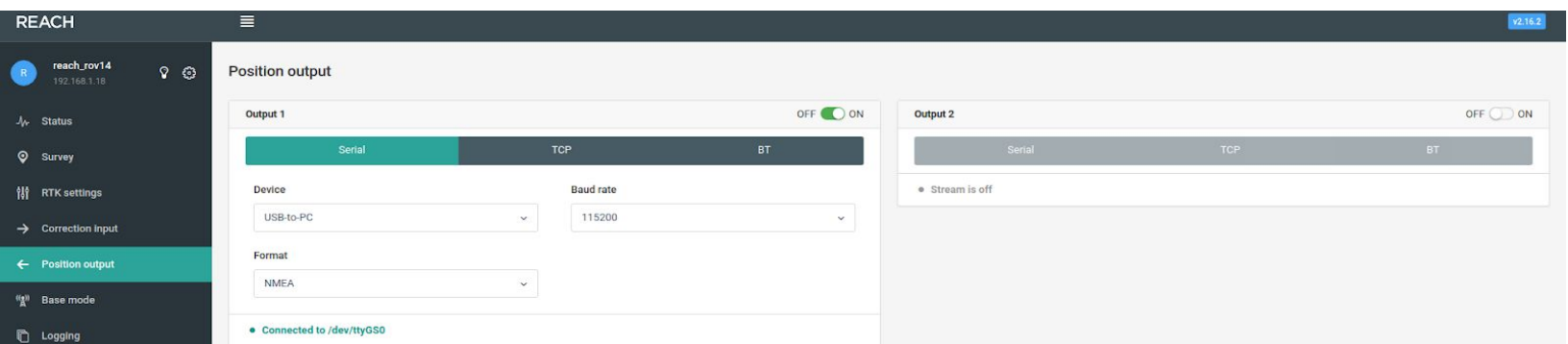

3. In the **Position output tab**, configure your rover to send data to your laptop through USB cable. The baud rate doesn't have to be 115200, but remember this setting as you are going to need it when reading data through serial in your code.
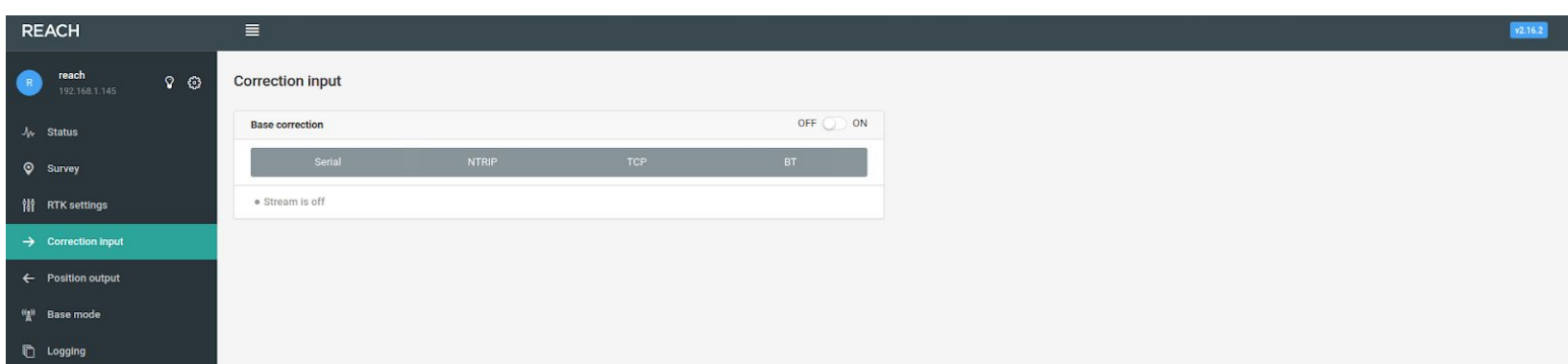
4. Turn off the base mode on the rover.

5. You may consider turning off the camera trigger and Bluetooth respectively in Camera **control tab** and **Bluetooth tab** to save some power.
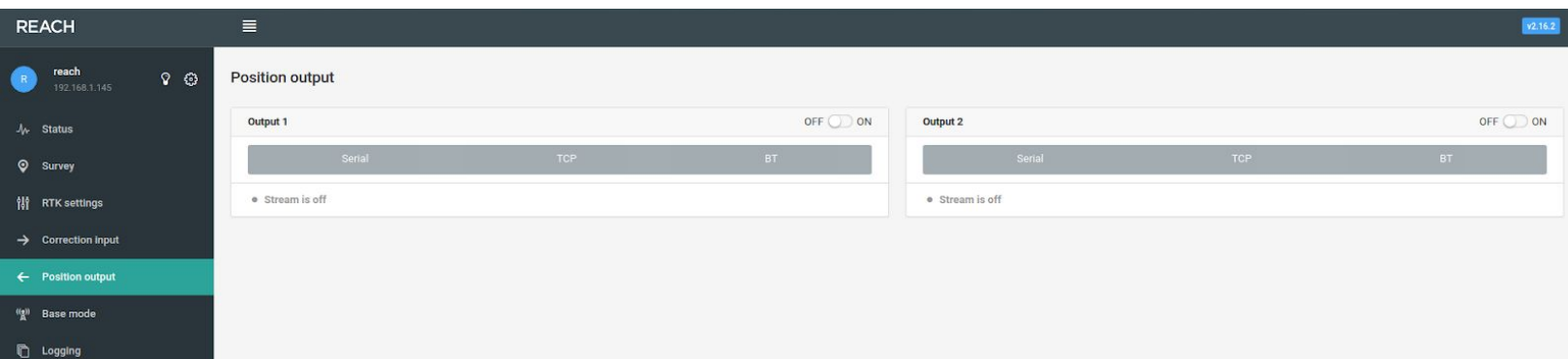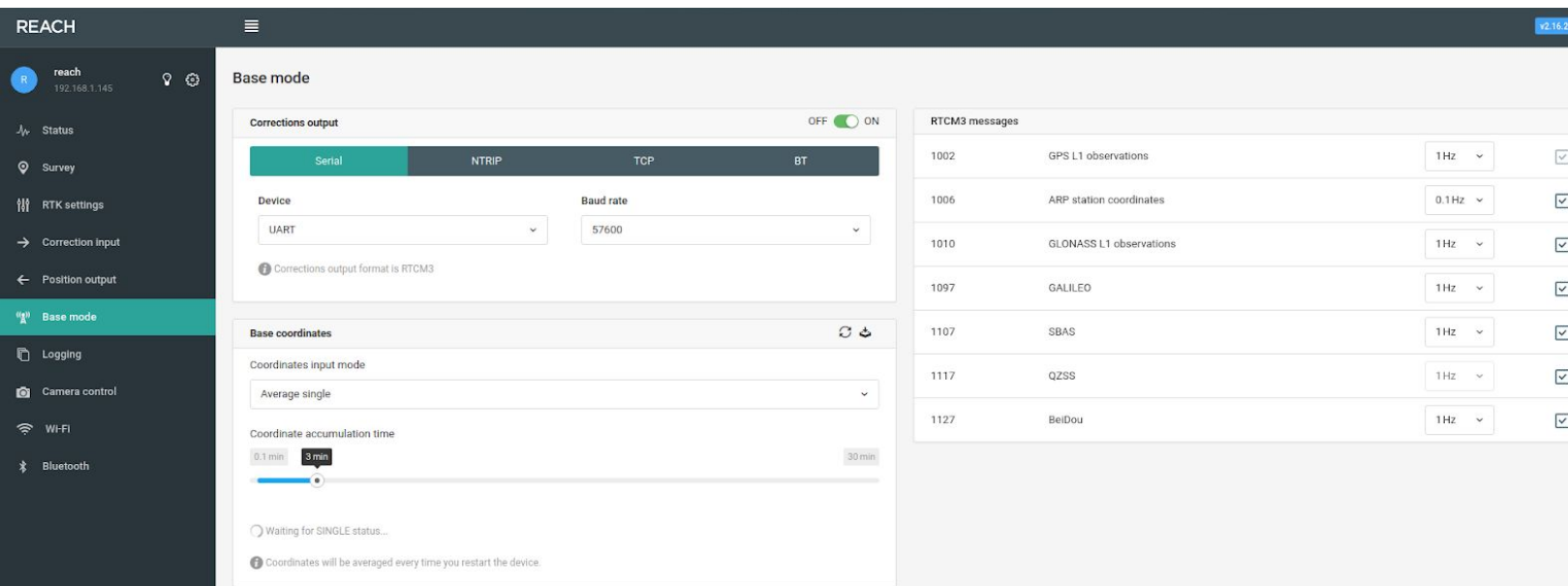
Appendix B.2
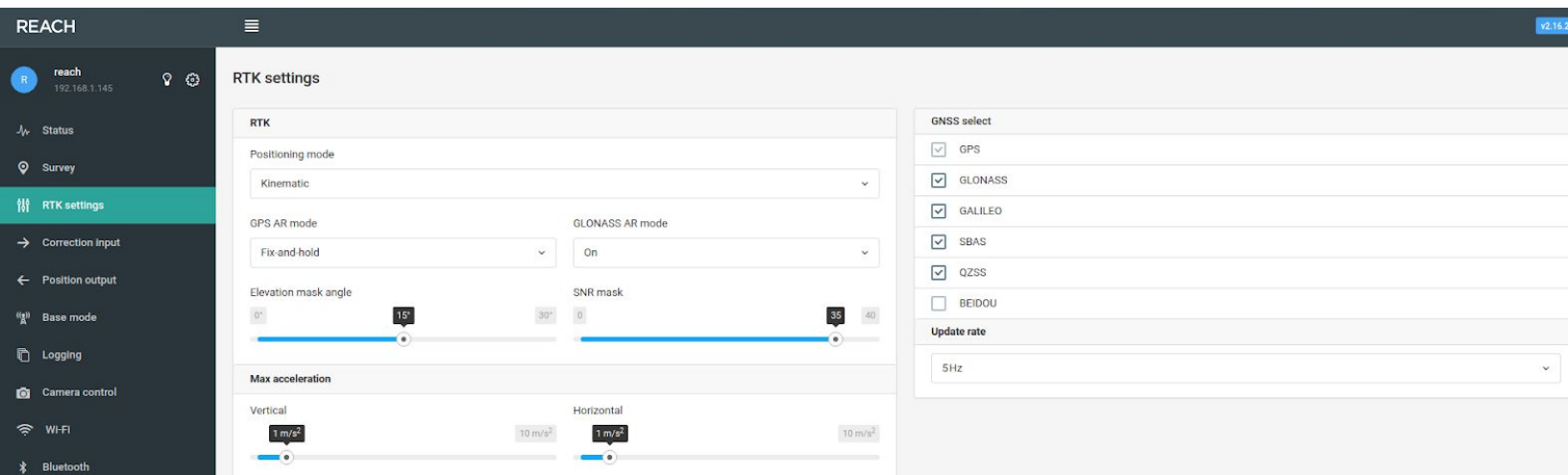
1. RTK settings on base side follow the same rules of those on rover side.

2. Rtk base will send correction information to rovers, so it doesn't take any correction input.

3. We are not taking position data from rtk base, so turn position output off as well.

4. The rtk base sends correction output to the radio through UART. Since the working baud rate of the radio is 57600, the baud rate of the rtk base should remain 57600, too. Coordinate accumulation time determines how much time the rtk base takes to calculate its current position. I recommend setting it to around 5 minutes, which should be sufficient for this lab.

Appendix C : Ublox ZED c009-f9p setup

Google_drive_link
https://drive.google.com/drive/folders/1WestLHb6dcniSqkmpvPclH7zgbHMIo94?usp=sharing

Above link has resources for setting up the zed f9p module and odin module

Flashing ODIN (common for both base and rover)

You need to flash the odin chip on ublox board to function as wifi hotspot properly and respond to AT commands.
1. Login to your windows machines
2. Install STM32 flasher from google drive link

Make sure your board doesn't have any jumper pins on uart ports. (very important)

steps
1. Download stm32 flasher zip folder from google drive, it has firmware files for odin inside.
2. Now open COMMAND PROMPT with admin privileges in windows.
3. Navigate to stm32 flasher directory using $ cd
4. Be very careful in typing or copy pasting following commands as you are giving memory address to flash tool, doing so in correctly might brick the device.
5. Check odin manual, flashing odin section 7 from page number 32-35
6. Keep device in safeboot mode
7. Flash mbed firmware
8. Restart your device and check for status led
9. Flash odin bootloader
10. Restart your device and check for status led
11. Flash odin w2 firmware
12. Open s- center software, make sure it enter AT mode.
13. Repeat steps 1-10 on the second device as well.

Configuring in Software
1. Open s-center software and follow this odins guide section 6 from pg 22-25
2. Download the config files for hotspot mode for both rover and base fromthe folder odin-w2 in the google drive.
3. Edit the config files by changing the wifi hotspot ssid name to 'UBX_WIFI_<x>'. Here <x> is the number with a marker written on your device.
4. Now configure your base,rover device with modified hotspot config files.
5. Restart your devices.
6. They should pair automatically, status led will indicate the same.

Adding Jumper pin

Now add the jumper pin in the location 3 as shown in the Fig 19 and fig 6 of odin user manual. This jumper connects uart x on odin to uart y on ZED F9P board, so corrections can be sent /received over wifi network. While configuring UBLOX zed F9P for RTK settings, REMOVE THIS JUMPER.

Zed f9p

This youtube video https://www.youtube.com/watch?v=FpkUXmM7mrc shows video instructions on how to set up. The uart ports, baud rates we will use are slightly different from the ones shown in the video. In the video he uses fixed mode, you have to use survey in mode as mentioned in pdf below.

In the google drive/c009-f9p eval board/config/C99 quick configuration.pdf has the same instructions as well.

In a nutshell, install u-center.exe on your windows machine, connect your eval board with usb, update your config file in generation 9 config view, update uart port settings in configuration view.