Course notes, week 12

Flight controller, Telemetry link

Kjeld Jensen kjen@mmmi.sdu.dk & Frederik Mazur Andersen fm@mmmi.sdu.dk

1 Agenda

1. Practical information
2. Presentation of week 11 lab exercises by team 4.
3. Introduction to the module exercises.
4. Exercises.

2 Theory presented in class

After the first 4 modules with mainly theoretical content, this module will be of a much more practical nature. As such there will be no formal theoretical lecture, however if problems arise during the experimental work we may address those in class.

In addition to the experimental work described below you are requested to read the publication 2018 *A survey of Open-Source UAV flight controllers and flight simulators* provided in the zip archive. This will provide you an overview of the current state of the art concerning open source flight controllers.

Each team will be issued the parts necessary for constructing a quadrotor drone. The experiments today will focus on the flight controller and the telemetry link.

There will be no report requirement for this module. Instead you are highly encouraged to spend the available time experimenting with each of the exercises to get the most out of them. Several modules later in this course as well as the RMUASD course taught in the Fall 2019 depend on knowledge and experience gained from these exercises.

1. UAS architecture
2. Flight controllers
3. Telemetry links

3 Exercises

3.1 Pixhawk flight controller

This section will setup and configure the flight controller.

3.1.1 Mount GPS on flight controller

Mount the GPS with tape on the flight controller. It should be mounted on top of the cube, on the flight controller. Make sure the arrow on the GPS module points at the front direction of the flight controller like shown in fig 1. There is an arrow on the flight controllers cube showing the front direction.

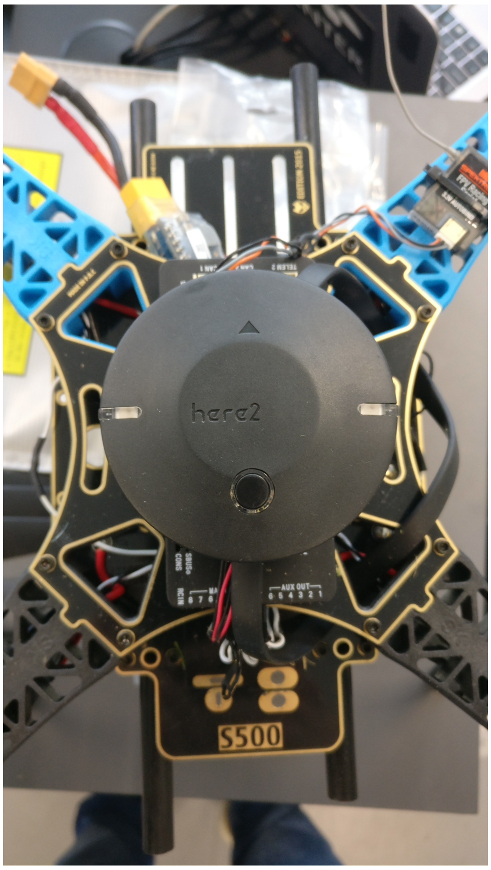


Figure 1: Image showing how the GPS module is mounted on top of the flight controller with the arrow in the front direction.

3.1.2 Installation of QGroundcontrol

Install QGroundControl to configure and communicate with the flight controller. Instructions to install is available at https://docs.qgroundcontrol.com/en/getting_started/download_and_install.html.

3.1.3 Flash PX4 stack to Pixhawk flight controller

Launch QGroundControl and follow the guide at https://docs.qgroundcontrol.com/en/SetupView/Firmware.html. Flash a stable version of the newest PX4 stack.

3.1.4 Changing parameters

1. Configure your flight controller to be a hexacopter in x-configuration. A guide to change airframe can be found at https://docs.qgroundcontrol.com/en/SetupView/airframe_px4.html.
2. Setup flight modes so that you have MANUAL, POSITION HOLD and MISSION on channel 6. Setup kill switch so it is active with channel 5. Guide is found at https://docs.px4.io/en/config/flight_mode.html
3. Calibrate the voltage sensor. Guide is found at https://docs.qgroundcontrol.com/en/SetupView/Power.html
4. Setup failsafes so that loss of RC signal and low battery errors make the drone land at its current position. Documentation is at https://docs.px4.io/en/config/safety.html

3.1.5 Sensor calibration

Calibrate all sensors. The documentation for calibration is found at https://docs.px4.io/en/config/. Compass, Gyroscope, Accelerometer and level horizon should be calibrated. Please notice that if you experience the error "mag-sensors inconsistent" after calibration, this is probably due to the GPS not being fixed to the flight controller, and thus the orientation of the two may differ.

3.1.6 Route plans

Create a mission in QGroundControl and upload it to the flight controller. After upload try downloading it from the flight controller and check if it matches with what was uploaded. Documentation is found at https://docs.px4.io/en/flying/missions.html

3.1.7 Mavlink LoRa via ROS

Mavlink LoRa is a MavLink communications library developed by SDU. Whereas the widely used MavLink library and associated ROS implementation have been developed with maximum flexibility in mind, the MavLink LoRa has been developed specifically for long range flights and minimal computing requirements.

1. Install ROS if you don't have it. (Kinetic for 16.04, Melodic for 18.04). Follow guide at: http://wiki.ros.org/melodic/Installation/Ubuntu

2. create a catkin workspace. Guide at http://wiki.ros.org/catkin/Tutorials/create_a_workspace

3. Download mavlink_lora from materials. Place the folder: <mavlink_lora/> in your <catkin_workspace/src>.

   Your structure should look like: <catkin_workspace/src/mavlink_lora/CMakeLists.txt>.

4. Open terminal and "cd" to your <catkin_workspace> and run "catkin_make"

5. Source your workspace with: "source devel/setup.bash"

6. Start mavlink_lora with "roslaunch mavlink_lora mavlink_lora.launch serial_device:=/dev/ttyUSB0 serial_baudrate:=57600 heartbeats:=true"

   (default port is "ttyUSB0" and "57600" baudrate, you might have to change that to "ttyACM0" and "115200" baudrate if using direct usb cable.)

   heartbeats:=true is nessecary to make mavlink_lora output heartbeats like a normal GCS would do. This is required as certain messages won't be sent from the flight controller unless it has a GCS connected. In a normal application you would have a GCS sending heartbeats to mavlink_lora that then relays them to the flight controller.

7. Run example python script <mavlink_lora/scripts/show_pos.py> and see you get the current GPS position out. (GPS position will probably not be stable as we are indoors.

3.2 Telemetry link

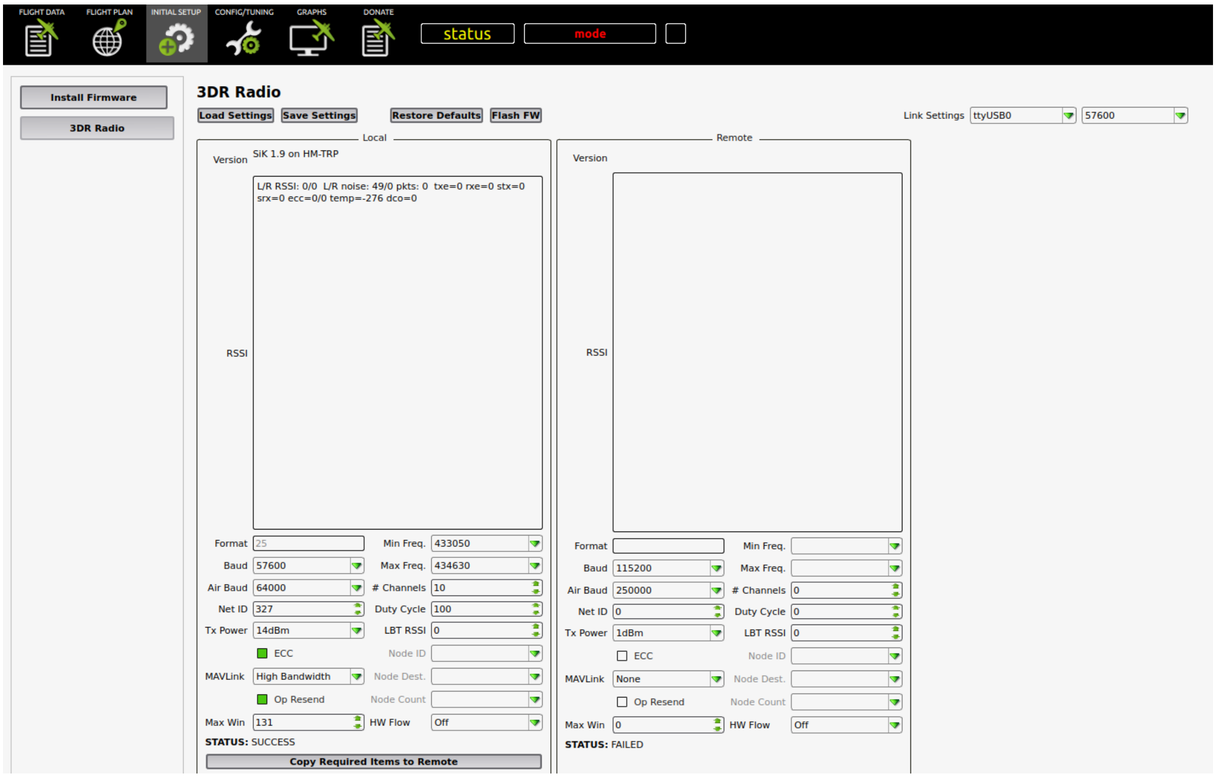
In this part we are setting up the telemetry links so the drone can be configured and communicated with without having a direct cable. NOTE: Please be careful with the telemetry module connectors, as they are quite fragile. Also please ensure that the antenna is attached before powering any modules.

3.2.1 Configuration

Before connecting the radio to the flight controller and the computer, the modules needs to be configured. This is done with APM planner. To install APM Planner follow instructions for your platform here http://ardupilot.org/planner2/docs/installing-apm-planner-2.html.

When APM planner is installed launch it and go to "initial setup" and then "3DR Radio" as shown in figure 2. Choose the right USB port and baud rate and click "load settings". The common baud rate is 57600, but the default baudrate for factory-reset modules is 115200.

If you experience issues with loading the settings try to open "Adv. Settings" on the right side and change the baud rate in there to 57600.



3DR Radio

Link Settings ttyUSB0 57600

Local

Version SiK 1.9 on HM-TRP

L/R RSSI: 0/0 L/R noise: 49/0 pkts: 0 txe=0 rxe=0 stx=0
srx=0 ecc=0/0 temp=-276 dco=0

RSSI

Format 25
Baud 57600
Air Baud 64000
Net ID 327
Tx Power 14dBm
☑ ECC
MAVLink High Bandwidth
☑ Op Resend
Max Win 131
Min Freq. 433050
Max Freq. 434630
Channels 10
Duty Cycle 100
LBT RSSI 0
Node ID
Node Dest.
Node Count
HW Flow Off

STATUS: SUCCESS

Remote

Version

RSSI

Format
Baud 115200
Air Baud 250000
Net ID 0
Tx Power 1dBm
□ ECC
MAVLink None
□ Op Resend
Max Win 0
Min Freq.
Max Freq.
Channels 0
Duty Cycle 0
LBT RSSI 0
Node ID
Node Dest.
Node Count
HW Flow Off

STATUS: FAILED

Figure 2: Configuring SIK modules though APM Planner.

When settings are loaded they can be changed and saved by clicking "save settings". Change the following settings and verify they got changed by loading settings again. The same settings have to be set on both modules.

The documentation for the settings can be found at http://ardupilot.org/copter/docs/common-3dr-radio-advanced-configuration-and-technical-information.html.

- Baud should be 57600. (Required by flight controller to talk with its module)
- Air Baud should be 64000
- Set Net Id to something unique that none of the other groups are using
- Tx power should be 12.5mW
- Set min. and max. frequency to fit the 433 MHz band.
- Mavlink should be on "high bandwidth"
- Max. Window should be 33
- ECC and Op Resend should be enabled
- Duty cycle should be 100
- Channels should be 20
- LBT RSSI should be zero
- HW Flow should be zero

Remember that both radios should have the same settings.

After configuring both radios, connect one of them to the flight controller on the TELEM1 port and the other to the computer. Launch QGroundControl and see that you got connection over the telemetry link.

3.2.2 Range tests

Do a range test. Start mavlink_lora with the telemetry module connected to a usb port, as was done previously. Run the python script <mavlink_lora/scripts/print_rssi.py>. You should now see RSSI and remote RSSI (remrssi) being output. It is output in dBm.

To see the effect more clearly install the python package "bokeh" with "pip3 install bokeh". Then run the script "python3 show_rssi.py" in the <mavlink_lora/scripts/> folder. This should open a browser window with a live updating graph showing the current RSSI.

Try to change the distance between the antennas or put obstacles in their way and see how the rssi behaves. Try to change the orientation of the antennas to see the effect of the polarization. You can also hold a wire behind the antenna to see the effect of reflectors. Play around with it and see how the theory works in practice.