**A4 – RB5 Drone Setup and Configuration**

This appendix contains instructions for safely setting up and flying the RB5 drone, both manually and programmatically, using ROS. A working level of experience with ROS and Ubuntu is assumed.

References:

<https://docs.modalai.com/> - manufacturer documentation

<https://forum.modalai.com/> - manufacturer forums

<http://wiki.ros.org/mavros> - general info on MAVROS

<https://docs.px4.io/main/en/> - PX4 documentation

There are 3 overall steps to get the drone working out of the box to flying with ROS:

* OS/firmware setup
* PX4 and Q Ground Control (QGC)
* ROS setup

A computer with Ubuntu 18.04+ is required.USB connection between the computer and RB5 drone may be easier with a computer that is natively running Ubuntu. However, it is still possible with a virtual machine (VM).

A4.1 – OS and Firmware

Steps 1 to 5 cover updating the drone’s operating system.

Steps 1 to 3: <https://docs.modalai.com/Qualcomm-Flight-RB5-system-image/#voxl-sdk-for-rb5-flight>

Steps 4 to 5: <https://docs.modalai.com/Qualcomm-Flight-RB5-voxl-sdk-upgrade-guide/>

1. Plug in RB5 Drone to computer and ensure it is detected. You can run **adb devices** and should see it. If you cannot see it, configure the VM or plug it into a machine that natively runs Ubuntu.
2. Go to <https://developer.modalai.com/> to find the proper system image. It is in *protected downloads* 🡪*RB5 Flight with VOXL SDK Platform Releases*. Download that file and unzip it using the command:
   * tar -xzvf {file\_name}

Graphical user interface, text, application

Description automatically generated

1. While the RB5 drone is connected to the computer, cd into the folder with the unzipped system image and execute:
   * sudo ./full-flash.sh
2. Back up camera calibration. Create a directory to store these files locally on your machine and execute adb pull to save them on your machine:
   * mkdir ~/cam\_cal\_files
   * adb pull /data/modalai/opencv\_tracking\_intrinsics.yml
   * adb pull /data/modalai/opencv\_stereo\_intrinsics.yml
   * adb pull /data/modalai/opencv\_stereo\_rear\_intrinsics.yml
   * adb pull /data/modalai/opencv\_stereo\_extrinsics.yml
   * adb pull /data/modalai/opencv\_stereo\_rear\_extrinsics.yml
3. On the host computer, navigate to where you downloaded the system image and install it by executing the following commands:
   * cd rb5\_platform\_1.3.1-0.8
   * ./install.sh
4. Push backup camera calibration files after installation is complete:
   * cd ~/cam\_cal\_files
   * adb push opencv\_tracking\_intrinsics.yml /data/modalai
   * adb push opencv\_stereo\_intrinsics.yml /data/modalai
   * adb push opencv\_stereo\_rear\_intrinsics.yml /data/modalai
   * adb push opencv\_stereo\_extrinsics.yml /data/modalai
   * adb push opencv\_stereo\_rear\_extrinsics.yml /data/modalai

A4.2 – PX4 and QCG

Before starting the steps in A3.2, download Q Ground Control (QGC).

Steps 1 to 5 enable wifi connectivity with the drone: <https://docs.modalai.com/voxl-wifi/>

Steps 6 to 11 enable MAVLINK and QCG communication with the computer

1. Setup a wifi hotspot. For example, you can use a cellphone and set SSID = “hotspot123” and password = “hotspot123”.
2. Connect the drone to the host computer via a USB cable, ensuring it is detected.
3. Open a terminal on the computer and execute the command:
   * voxl-wifi station <ssid> [password]
4. The next time the drone is rebooted, it should automatically connect to that network. Connecting to the drone via ssh on the host computer should be possible.
5. To connect to the drone via ssh, ensure both the computer and drone are on the same Wi-Fi network. On the computer, open a terminal and execute:
   * ssh root@{DRONE\_IP}
     + password is *oelinux123*
     + DRONE\_IP is found by going on connections 🡪 mobile hotspot and tethering 🡪 mobile hotspot 🡪 connected devices
6. Configure mavlink by executing the following command and following the prompts
   * voxl-configure-mavlink-server

Text

Description automatically generated

1. Enable px4 and setup QCG IP by executing the following command and responding to the prompts as follows:
   * voxl-configure-vision-px4

Text

Description automatically generated

QGC IP address is the IP address of the host computer. It can be found using the command:

* ifconfig -a

1. Update the .param file on QGC. It can be found at the following github repo: <https://github.com/albud187/RB5_MAVROS_params> as the filename “RB5\_MAVROS\_VIO\_09-26.params”
2. Open QGC 🡪 vehicle setup 🡪 parameters 🡪 tools 🡪load from file.
3. This .param file is required to allow the drone to arm and fly **without GPS.**
4. The drone can now be flown safely without GPS using **position mode** using the virtual joystick. If not, wait for this popup to appear on QGC, indicating that VIO has started.

Shape, rectangle

Description automatically generated

After this popup, the drone should be able to switch to position mode. Position mode requires the drone to have a pose estimate, such as from VIO.

Map

Description automatically generated

A4.3 – ROS setup

Steps 1 to 7: <https://docs.modalai.com/setup-ros-on-voxl-2/>

Steps 8 to 14: <https://docs.modalai.com/mavros-voxl-2/>

1. Open a terminal and ssh into the drone.
2. Execute:
   * sudo apt-get update
   * sudo apt install -y ros-melodic-ros-base ros-melodic-image-transport
   * voxl-configure-mpa -p -f rb5-flight
3. Open ~/.bashrc by executing:
   * nano ~/.bashrc
4. On ~/.bashrc add the following line:
   * . /opt/ros/melodic/setup.sh
5. Save and close the ~/.bashrc file. Then source it by executing:
   * source ~/.bashrc
6. Now, you can execute:
   * roslaunch voxl\_mpa\_to\_ros voxl\_mpa\_to\_ros.launch
7. The command in step 6) will start publishing the camera imagery as ROS topics.
8. Install MAVROS by executing:
   * Sudo apt-get install ros-melodic-mavros ros-melodic-mavros-extras ros-melodic-control-toolbox
   * cd /opt/ros/melodic/lib/mavros
   * ./install\_geographiclib\_datasets.sh
9. Build mavros test by executing:
   * cd /home
   * git clone -b simple-example <https://gitlab.com/voxl-public/support/mavros_test.git>
   * cd /home/mavros\_test
   * ./build.sh
10. Configure ROS environment. Open ros\_environment.sh by executing:
    * nano /home/mavros\_test/ros\_environment.sh
11. Under configure IPs, set them to the following:
    * export ROS\_MASTER\_IP=localhost
    * export ROS\_IP=localhost
    * export ROS\_MASTER\_URI=http://localhost:11311/
12. Source the ROS environment on the ~/.bashrc so you do not have to source it each time. Open the .bashrc file by executing:
    * nano ~/.bashrc
13. Add the following line to the ~/.bashrc:
    * source /home/mavros\_test/ros\_environment.sh
14. Now you can launch MAVROS by executing:
    * roslaunch mavros px4.launch fcu\_url:=udp://127.0.0.1:14551@:14551 tgt\_system:=${PX4\_SYS\_ID}