KOMPAÏ robotics

High level software overview for USC's robulab10

From :

To :

Date : 02/12/2022

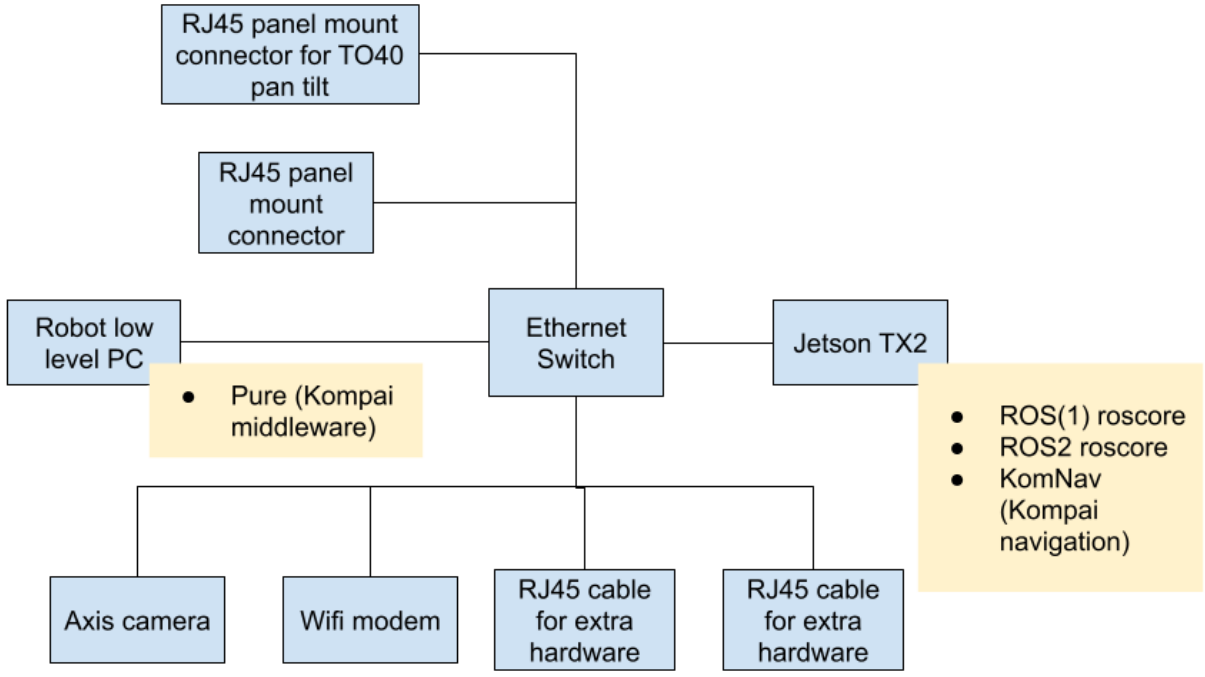
Object :

This document specifies the software architecture inside the robot's onboard Jetson TX2 module. Kompai provides a ROS interface with the robot's low level controller (angular and linear speed control of the robot, laser data, odometry data, IR sensors), the onboard ethernet camera, the TO40 pan tilt platform (not part of the robot). Kompai also provides a high level navigation program (KomNav), for localisation and navigation of the robot, it has an HTTP API.

A crontab automatically starts all the necessary programs after the system has booted.

KOMPAÏ robotics

1. Robot's high level internal hardware architecture :



The main modules communicate on the LAN ethernet network. The robot's low level PC is also interfaced with the lower level sensors and actuators (motor drivers, laser,...).

The following devices have the following IP addresses on the LAN network:

Device	IP address
Axis camera	192.168.1.10
Wifi modem	192.168.1.1
Jetson TX2	192.168.1.30
Low level PC	192.168.1.2

Because there is no DHCP server on the network, each device needs to have a static IP.

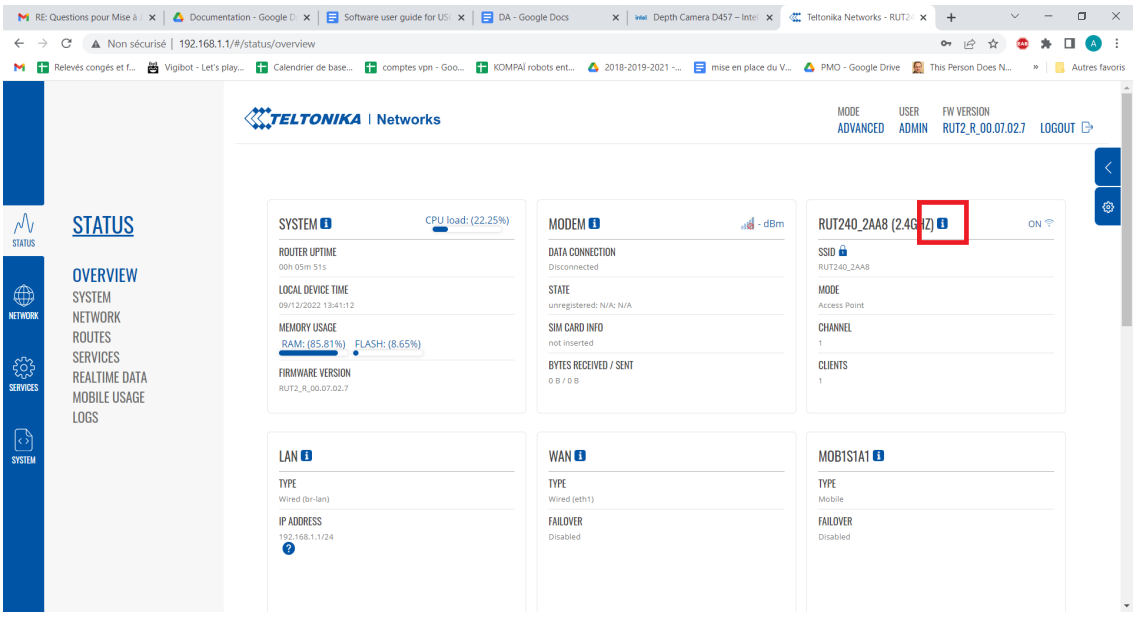
2. Wireless modem

This device is the entry point to the robot when accessing it wirelessly. It creates its own wifi network to which you can connect. *SSID: RUT240_2AA8, password: b4NVa7e9.*

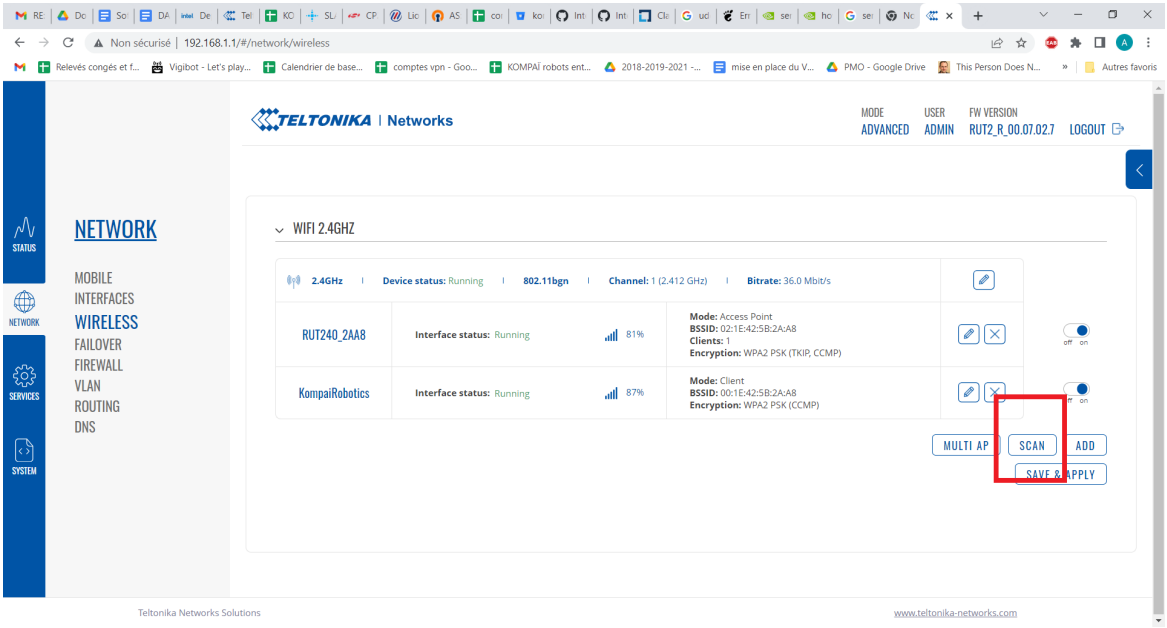
It can also connect to an existing wifi network for internet access. This needs to be configured from the modem's web based interface, accessible at 192.168.1.1.

To connect the modem to an existing wifi network, first connect to the wifi network created by the robot, details given above. Then follow these steps:

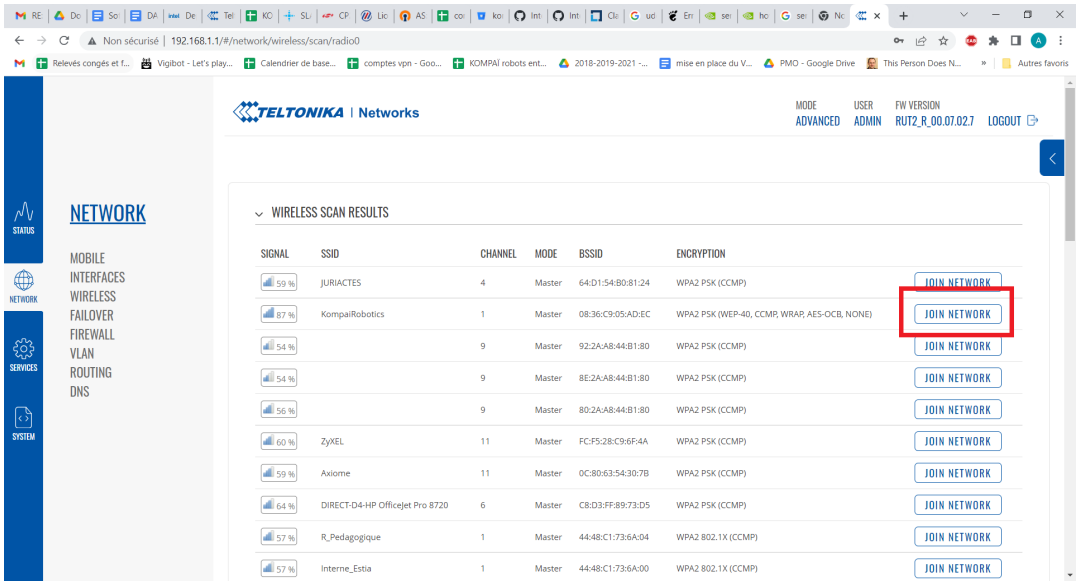
1. Open a web browser and type 192.168.1.1 in the address bar
2. Use login: "*admin*" and password: "*Admin001*" to login
3. Click on the information button next to RUT240_2AA8 (2.4Ghz)



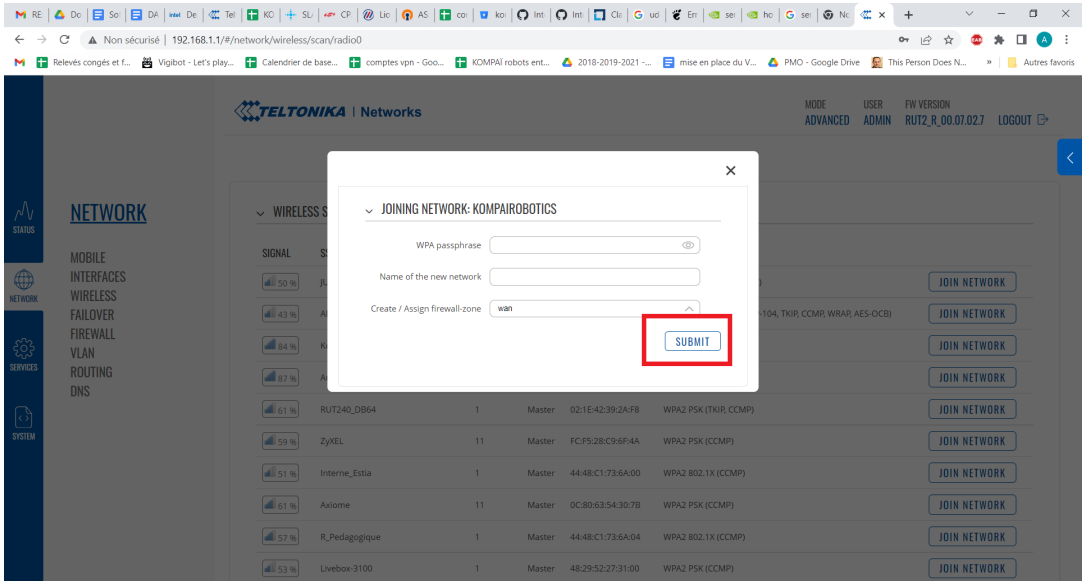
4. Click on scan



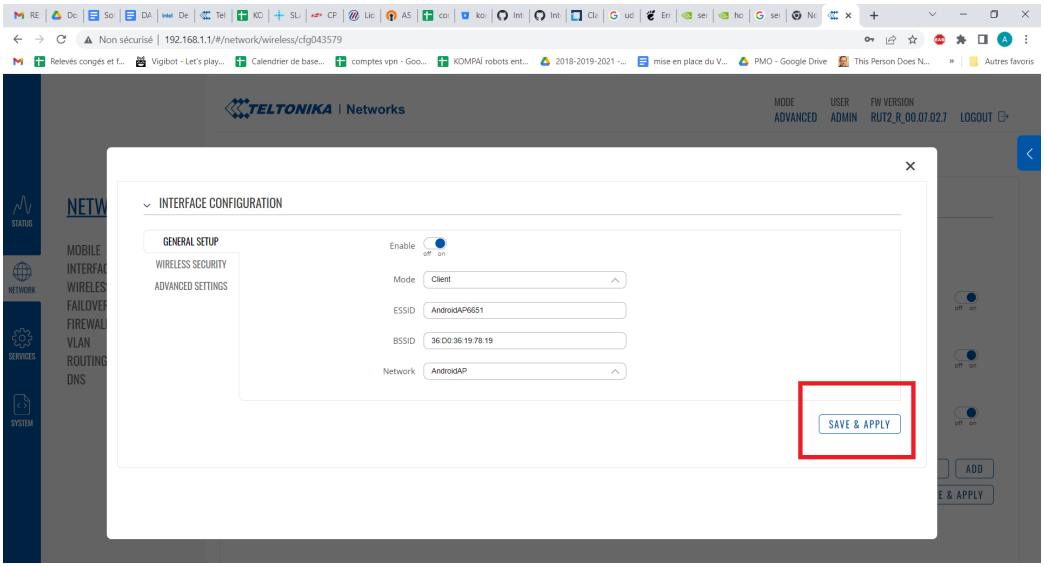
5. Choose the network you want to connect to and click on Joint Network



6. Input the network password, give it a name and click on Submit



7. Finally click on Save & Apply



You are done! It can take a little bit of time for the modem to connect to the network and reboot of the robot may or may not be necessary.

3. Nvidia Jetson TX2 :

The jetson is running Ubuntu 18.04, it auto-logins with user “nvidia”, for which the password is “nvidia” as well. SSH and VNC access are both installed.

4. Autostart :

The above crontab entry starts the file /home/nvidia/Desktop/Autostart/start_robot.sh at boot. It then calls the different ROS interfaces and finally starts KomNav.

The main startup file (start_robot.sh) calls several subsequent bash scripts; their command logs output are located at /home/nvidia/Desktop/Logs and can be useful for troubleshooting should any problems occur.

The start_ros.sh file located in /home/nvidia/Desktop/Autostart/USC is responsible for setting up the ROS environment variables and launching the main launch file.

The start_komnav.sh file located in /home/nvidia/Desktop/Autostart/USC starts komNAV. By default it is commented, since it can’t be started at the same time as the ROS PURE bridge.

The start_ros2.sh file will do the same thing for the ROS 2 interface. Starting one or the other is a matter of un/commenting the right lines in the /home/nvidia/Desktop/Autostart/USC/autostart.sh.

The start_komnav.sh files start proprietary Kompai software and should not be modified unless explicitly specified by Kompai.

Sumup:

Base file location: /home/nvidia/Desktop/Autostart

5. ROS(1) (melodic) :

The following ros topics are available on the robot:

Designation	Interface name	Interface type	Message type	Comment
Linear and angular command	/cmd_vel	Subscriber	geometry_msgs::Twist	
Lidar data	/scan	Publisher	sensor_msgs::LaserScan	
Odometry data	/vel	Publisher	geometry_msgs::Twist	
Pan-tilt camera control	/pan_command /tilt_command /left_command /right_command	Subscriber	pure_ros::MotorCommand	Custom message
Pan-tilt camera state	/pan_state /tilt_state /left_state /right_state	Publisher	pure_ros::MotorState	Custom message
IR sensor	/ir_range	Publisher	sensor_msgs/Range	One topic per sensor
Axis camera	/front_camera/image_raw	Publisher	sensor_msgs/Image	Per ROS package http://wiki.ros.org/axis_camera

The camera image is created using the existing ROS axis_camera package, the launch file for it is located at /home/nvidia/ROS/robot_ws/src/kompai_ros/launch.

The PURE ROS bridge source is in the package located at /home/nvidia/ROS/robot_ws/src/pure-ros/.

KOMPAÏ robotics

6. ROS2 :