CODE REVIEW v1.0 (Functional)

Microprocessor Code: Jetson Nano

Directory Structure:

avia_pkg

- **|----- arduino**
  - **|----- node_arduino1**
  - **|----- node_arduino2**
- **|----- launch**
  - **|----- main.launch**
- **|----- recipe**
  - **|----- dunzo.json**
- **|----- schema**
  - **|----- OppoA54.json**
  - **|----- SamsungJ6.json**
  - **|----- XaiomeNote11.json**
- **|----- script**
  - **|----- json_engine.py**
  - **|----- node_recipe_control.py**
- **|----- src**
- **|----- auto_launch.sh**
- **|----- CMakeLists.txt**
- **|----- main.py**
- **|----- package.xml**

auto_launch.sh

A. Brings up the ethernet network with a fixed static IP address 192.168.1.100 (comment: eth_0 up)
B. Sourcing the bash files required for
   a. /opt/ros/melodic/setup.bash - To initialize ROS Melodic
   b. ~/avia_ws/devel/setup.bash - To locate the files/packages in the ROS workspace
C. Executes the ROS file: *main.launch*

main.launch

A. Runs the *main.py* program
B. Starts ROS Serial communication with Arduino

main.py

MQTT Topics(6)

- configure_wifi
- feedback_wifi
- control_leds
- control_fans
- control_cu
- launch_file

ROS Topics(3)

- control_leds
- control_fans
- control_cu

node_recipe_control.py

Execute the one by one steps in the recipe file

ROS Topics(3)

- control_stepper
- control_tapper
- feedback_stepper_state

json_engine.py

Define the schema variable from the schema file required for the operation

node_arduino1

Control the movement of all stepper motors and tapper

ROS Topics(3)

- control_stepper
- control_tapper
- feedback_stepper_state

node_arduino2

Control the leds, fans, cooling unit and read the limit switch state

ROS Topics(7)

- control_leds
- control_fans
- control_cu
- feedback_switchX_state
- feedback_switchY_state
- feedback_switchZ_state
- feedback_switchTray_state