For starting up the SSYX02-16-27 project. 5 UWB RCMs, 3 laptops and 2 robots are needed.

For plotting reasons it is recommended to use 2 robots only since there are hardcoded parts for the plotting. If plotting is of no interest and multiple robots are wanted please comment out the plotting function in MainController.py. This startup guide will also presume that the end node is a fixed point but code support for a RCM for end node is included (out commentated as of now).

Setting up a whole new system:

1. Set up a grid system in a test site for which on three anchor RCMs are placed.
2. Place three RCMs at fixed positions that are known and connect power supply.
3. For each robot:
    a. Connect a laptop to a robot via serial-to-usb
    b. Connect RCM to laptop and configure the RCM after the RCM SETUP guide.
    c. Install ROS after the guides from http://wiki.ros.org/ROS/Installation
    d. Download code from GITHUB-repo
    'git clone https://github.com/mightoz/SSYX02-16-27'
    e. Install ROSARIA into the rosaria directory of SSYX02-16-27/robot_ws: after the guides from http://wiki.ros.org/ROSARIA/Tutorials/How%20to%20use%20ROSARIA
    f. Catkin make from robot_ws and ensure that the code is buildable
    g. Add the following to .bashrc(make sure that the paths are correct for where the downloaded code was put):
    source /opt/ros/indigo/setup.bash
    source ~/SSYX02-16-27/robot_ws/devel/setup.bash
    h. In SSYX02-16-27/robot_ws/src/robotclient/src/Measure.py, change the positions of the Anchor RCMs to where they

were set up. Make sure that this is the same for **every** laptop(for example change one and git push/git pull the new positions)

4. For base station:
    a. Download ROS and Code similarly to the robot
    b. Connect a RCM for base station positions.
    c. Make sure that the positions of the anchor is correct in Measure.py for the base station aswell.
5. Set up the network after the NETWORK STARTUP/BUG guide.
6. The following should be added into .bashrc for each laptop(robot and base):
   export ROS_MASTER_URI=http://192.168.2.1:11311/ %This should be the IP-address of the AP.
   export ROS_IP=192.168.2.1 %This should be the ip of the laptop
   export ROS_HOSTNAME=base %This should be the hostname of the laptop
7. It might be a good idea to add the host names and ip addresses into each laptops hostfile to ensure that each robot is known. For example:
   192.168.2.1 Base
   192.168.2.2 robot1
   192.168.2.3 robot2
8. Now everything should be set up.

If using the/a already set up system:

1. Make sure that every laptop is updated with the latest code.
2. Make sure that the positions of the anchors is correct in Measure.py for each robot+ base station.
3. For robot:
    a. Connect laptop to robot via serial and to RCM via ethernet.
    b. Supply power to RCM (perhaps laptop as well if low on battery).
4. For Base:
    a. Connect RCM via ethernet + power supply.

b. Start up the AP on the base station with the command:
   sudo hostapd /etc/hostapd/hostapd.conf
c. If the response is: “nl80211: Could not configure driver mode” . Please see the NETWORK START UP/BUG guide for solution.
d. Run sudo ifconfig wlan3 192.168.2.1/24 for AP in new terminal.
e. Make sure that the clients are connected to the AP(ping)
f. Start a roscore with:
   roscore

5. For robot: (SSH*)
   a. Either run all services in one terminal finish each command with & (no prints) or run every command in a different terminal (information prints)
   b. Run the following commands, N is the robot number(each new line is a space)**:

      rosrun robotclient serviceRotate.py __name:=rotateRobot_serviceN rotateRobot:=rotateRobotN RosAria/cmd_vel:=robotN/cmd_vel

      rosrun robotclient serviceMoveForward.py __name:=moveRobot_serverN moveRobot:=moveRobotN RosAria/cmd_vel:=robotN/cmd_vel

      rosrun robotclient UpdateTwistService.py __name:=update_twist_server_N updateTwist:=updateTwistN RosAria/cmd_vel:=robotN/cmd_vel

      rosrun robotclient get_coord_server.py __name:=get_coord_serverN get_coord:=get_coordN _ip_of_uwb:=IP %This is the ip of the RCM, the three digit number displayed on ethernetport.

c. Start the robot.
d. Run the rosaria node with, N is the robot number:
   rosrun rosaria RosAria __name:=robotN***

6. For Base:
   a. Start a new terminal and run the following for the base position:

      rosrun robotclient get_coord_server.py __name:=get_coord_server0 get_coord:=get_coord0 _ip_of_uwb:=103 %This should be the ip for base RCM

   b. Start a new terminal and run:

      rosrun masterclient MainController.py _nbr:=4 %This should be the number of NODES used (=> Base + robot *N + End).

   c. (Optional if printouts are wanted: Start a new terminal)Run:

      rosrun masterclient MainControllerStartup.py _cordfunc:=align2 %where align2=simultaneous and align1=iterative. The iterative is only supported on the branch

   d. Now it's running! Ctrl-C on termination. The program terminates on command(to enable a movable base). ****
   e. Make sure to terminate the rosaria node manually on each robot to stop the motors fully.

*SSH is setup on the laptops currently used, simply SSH to multipos1@robot2 and multipos@robot1.

**For the laptops currently used, bash aliases are set up to simplify this, they are:
move1, twist1, rotate1, coord1 and respectively for robot number two.

Also rotate services are only necessary for the iterative coordination and can be skipped if simultaneous is to be run.

***For the laptop setup for robot2 currently used a re-chmod of the port for the rosaria node has to be done after a restart:
chmod a+rw /dev/ttyUSB0 before starting rosaria node again.

****Note that when SSH, the first run the rosaria node doesn't start the motors correctly so restart of the rosaria nodes could be necessary.