Instructions

Setting up the car

* The diagram of the hardware setup for the car is under the hardware folder of this repo
  + The Arduino Micro is in a breadboard, the VDD rail is tapped from the output of the buck converter. We had power problems otherwise.
  + The buck converter should have an output of 5V, it can be adjusted using the potentiometer on the board
* Make sure that ROS is installed on both the master computer and the microcontroller being used, <http://wiki.ros.org/indigo/Installation/Ubuntu> , also make sure to create a catkin workspace
* Install the packages that will be used on the Odroid
  + Joy\_node – Also needed on master computer
  + Gmapping
  + ROS Serial to Arduino
  + Urg\_node
  + Robot\_localization
  + Myahrs\_driver
* In the src folder, in the catkin\_ws, clone the git repo
* Run ‘catkin\_make’ and then ‘catkin\_make install’
* Install the Arduino IDE so you can program the Arduino
  + <https://www.arduino.cc/en/Main/Software> - the software
  + <http://wiki.ros.org/rosserial_arduino/Tutorials/Arduino%20IDE%20Setup> – ros setup
  + <http://wiki.ros.org/rosserial/Tutorials/Adding%20Other%20Messages> – follow this so the messages for the encoder and joy controller work
  + Remember to do “source ~/catkin\_ws/devel/setup.bash” in every new terminal window
  + Program the Arduino sketch

Running the ROS nodes

* The Odroid should be setup so when plugged in it automatically connects to the SACAR\_network
  + The login for the router is user: SACAR pass: SACAR
  + The password set for the SACAR\_network is ‘SACAR\_network’
* On the main computer
  + Run ‘roscore’
  + ‘export ROS\_MASTER\_URI=http://NAME:11311’ where NAME is replaced with the name of the device that is master, in the terminal the roscore was run it will have the full URI listed
  + ‘rosrun joy joy\_node’ to start the node for the controller to publish its’ commands to a topic
  + ‘ssh odroid@IP’, SSH into the odroid in a new terminal, if it cannot find the odroid check the host file under /etc/, the IP is normally 192.168.1.101
  + ‘export ROS\_MASTER\_URI=http://NAME:11311’, Tell the odroid who is the master, which is the same master as the first
  + ‘source /catkin\_ws/devel/setup.bash’, makes it possible to use ros commands
  + ‘roslaunch sacar main.launch’, to run the launcher for the nodes that are being run on the odroid
  + ‘roslaunch sacar ukf\_localization.launch’, Another launch for robot localization
  + ‘roslaunch sacar gmapping.launch’, Another launch for gmapping

Live Camera Feed

This was setup following to this link <https://trac.ffmpeg.org/wiki/ffserver>

* First install ffmpeg along with its’ dependencies
  + <https://trac.ffmpeg.org/wiki/CompilationGuide/Ubuntu>
  + The Audio dependencies can be left out when installing
* Place the ffserver.conf (server configuration file) in the /etc/ folder
* ‘ffserver –f /etc/ffserver.conf’ , The server is started with the command\*
* ‘ffmpeg –f v4l2 –s 640x480 –r 10 –i /dev/video0 <http://localhost:8090/feed1.ffm>’, The video feed is started with the command\*
* The feed can then be viewed from <http://IPADDRESS:8090/test.mjpg> where IPADDRESS is the ip of the device the camera is plugged into, normally 192.168.1.101

\*If the camera is mounted on the car and you want to start the camera feed through the computer you have to SSH into the Odroid, once for each command

Router

* Change the subnet of the router so it is 1 instead of 0 (ex: the IP would change from 192.168.0.101 to 192.168.1.101
* In the host file (in the /etc/ folder) make sure the IP matches the name of the device (this makes it possible to do ‘ssh NAME@IP’)
* Have the microcontroller auto-connect to network
* For the current router we have the login is user: SACAR pass: SACAR
  + 192.168.1.1 is the current address since we changed the subnet