Robot

# Git

* To push single branch to remote: (origin = Samana-...

|  |
| --- |
| git checkout feature\_x git push origin -u feature\_xg |

* + use -f if you need to overwrite backup branch
* Undo last commit:

|  |
| --- |
| git reset --soft HEAD~1 |

* To move commits from backup\_branch to other branch:
  + Go to new branch
  + Cherry pick commits you want
  + Delete backup\_branch
  + Create backup\_branch

# GPS

* find ip <http://whatsmyrouterip.com/> use your device IP append :11311/
* or in terminal write ***ip route*** and find last ip *scope link src* ***192.168.0.102*** *metric* append **:11311/**
* export ROS\_MASTER\_URI=[http://192.168.0.100:11311/](http://192.168.0.104:11311/)
* roscore
* Connect via ROS Sensors Driver on Android with same IP
* No error should be shown

# General programming

* IMPORTANT: sync clocks with ROS clock
* F() Macro on Arduino printing. To save SRAM
* nh.getHardware()->setBaud(1000000);
* rosrun rosserial\_arduino serial\_node.py \_baud:=115200 \_port:=/dev/ttyUSB0
* delay(1);
* Be careful with rosserial because it can desync if data is sent too fast
* 92% memory usage causes instability
* Lost sync with device, restarting…
  + Lower baud rate
* To build ROS library for Arduino

1. build samana\_msgs

|  |
| --- |
| source devel/setup.bash rm -r ~/Documents/SamanaAutonomousRobot/Arduino/libraries/ros\_lib rosrun rosserial\_arduino make\_libraries.py ~/Documents/SamanaAutonomousRobot/Arduino/libraries/ |

1. If can’t include headers first include ros.h
2. [wiki.ros.org/rosserial\_arduino/Tutorials/Arduino%20IDE%20Setup](http://wiki.ros.org/rosserial_arduino/Tutorials/Arduino%20IDE%20Setup)

* Further if adding customized ros\_lib for a node:
  + In Samana project main folder */Arduino* modify ***make\_links.sh*** to create a symbolic link of *ros\_lib\_newnode* to a Arduino/libraries folder on computer (place where IDE reads libraries) e.g.

|  |
| --- |
| ln -sv /home/combinacijus/Documents/SamanaAutonomousRobot/Arduino/libraries/ros\_lib\_odom/ /home/combinacijus/Documents/Arduino/libraries/ |

* If ROS workspace change or adding new

|  |
| --- |
| sudo subl .bashrc source "/home/combinacijus/Documents/SamanaAutonomousRobot/ROS/samana\_ws/setup.sh" |

* + .../build/CMakeCache.txt change all directories to new one
  + Also in RoboWare change settings.json for autocomplete
* If node tried to send before init “Tried to publish before configured”:
  + Build project (catkin\_make)
  + Source devel/setup.sh
  + Check rosmsg list if your package messages are loaded
* To accept arduino node to roscore

|  |
| --- |
| sudo chmod 666 /dev/ttyUSB\* rosrun rosserial\_arduino serial\_node.py \_baud:=10000000 \_port:=/dev/ttyUSB0 \_recal:=false |

* UDEV rule for separating Aruduino depending on devpath (usb port)

|  |
| --- |
| SUBSYSTEM=="tty", ATTRS{idVendor}=="1a86", ATTRS{idProduct}=="7523", SYMLINK+="ttyUSB-arduino%s{/devpath}" |

* + This gives names like: /dev/ttyUSB-arduino2.1 /dev/ttyUSB-arduino2.3.3

# General electronics:

* **IMPORTANT:** Capacitor on input to eliminate power loses

# IMU BNO055

* SCL: A5; SDA: A4, 5V, GND
* Human ride: max angular velocity 300deg/s
* Human ride: max acceleration 3m/s^2
* BNO055 loses calibration when space between hoverboard is <20cm (30cm from any electronic would be nice)
* On system calibration loss. 360 + 180 - 180 is advised to recalibrate IMU
* IMPORTANT: On magnetometer calibration degradation one wheelie to different headings might calibrate it
* **IMPORTANT:** After loosing power IMU won’t recover. Needs extra code. Problem with Wire.h library in twi.cpp file, issue: <https://github.com/panStamp/arduino_avr/issues/1>. Maybe needs extra Arduino to reset via pin.
* PROBLEM: Heading is not reliable
* Baud rate could be lowered to 57600 but it will reduce update speed from 100hz to 73hz

# Distance sensors

* Max distance 4-6m at home
* Need spacer in front so minimum distance wouldn’t be 0. Else it would cause bugs
* SRF04 will only listen to echo if it was triggered
* Ultrasonic sensors on other robots might trigger own sensors therefore giving shorter distance to the obstacle
* IMPORTANT: deal with bump sensors when angled
* avrdude: stk500\_recv(): programmer is not responding. But will upload via Arduino IDE
* IMPORTANT: ROS: check each sensor max distance. If it’s too big map won’t be accurate it will have wall at the end of sonar cone

# Hoverboard

* IMPORTANT: There is inactivity timeout on hoverboard firmware. It probably should be disabled for competition
* Main battery voltage screen shows **0.4V lower voltage**
* Battery:
* 3.6V Low voltage no beep
  + 3.5V Almost empty battery; beeps
  + 3.37V Turns offs when almost no speed
  + Charging speed. no\_load=35.1V to no\_load=38.6V in 1hr; to 41.6V 1h45m

# Arm

* IMPORTANT: if motors disables on their own it might be because overcurrent protection
* If motors don’t work all manual switches might be on

Times with no extra weight and grabber on lifter down position:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Dry thread | Lubed thread | Lubed long ago |
| Lifter up | 18s | 18s |  |
| Lifter down | 13s | 13s |  |
| Grabber open | 11s |  |  |
| Grabber close | 11s |  |  |

# Object recognition

* Pretrained YoloV3: <https://github.com/OlafenwaMoses/ImageAI/releases/tag/essential-v4>
* *You can keep the images at their original size. The training network will automatically re-scale the image to sizes between* ***288 to 488*** *as it runs through many experiments.*
* Too large images will exhaust GPU memory
* On CPU 0.34 - 0.42 FPS on 100%; on GPU

### Docker

|  |
| --- |
| sudo docker run -it --rm --gpus all --net host --device /dev/video2 -v /home/combinacijus/Documents/SamanaAutonomousRobot/Python/GoldBagDetector/:/notebooks/GoldBagDetector --name imageai combinacijus/imageai:compute3.0  sudo docker run -it --gpus all -v ~/Documents/SamanaAutonomousRobot/Python/GoldBagDetector/:/notebooks/GoldBagDetector imageai  sudo docker run --gpus all -it --rm tensorflow/tensorflow:1.12.0-gpu /bin/bash  sudo docker run --gpus all -it -v /host\_path:/container\_path tensorflow/tensorflow:1.12.0-gpu /bin/bash   sudo docker images  sudo docker ps -a  sudo docker start -ai CONTAINER  sudo docker commit CONTAINER NEW\_IMAGE\_NAME  sudo docker rmi IMAGE  sudo dokcer stop CONTAINER   sudo docker login  sudo docker login docker.io  sudo docker tag ID combinacijus/imageai  sudo docker push combinacijus/imageai |

# ROS

Python nodes might get **stuck** running after shutting down launch file and would eat a lot of resources in that case:

|  |
| --- |
| kill -9 <PID> |

To visualise range data (launch file):

* Make URDF model
* Find and Load URDF model to param server:

|  |
| --- |
| <arg name="model" default="$(find urdf\_tutorial)/urdf/01-myfirst.urdf"/> <param name="robot\_description" command="$(find xacro)/xacro.py $(arg model)" /> |

* For loading rviz config:

|  |
| --- |
| m<arg name="rvizconfig" default="$(find urdf\_tutorial)/rviz/urdf.rviz" /> |

* Publish joint states (all rotation of the robot etc):

|  |
| --- |
| <node name="joint\_state\_publisher" pkg="joint\_state\_publisher" type="joint\_state\_publisher" /> |

* Publish all TFs of the robot:

|  |
| --- |
| <node name="robot\_state\_publisher" pkg="robot\_state\_publisher" type="state\_publisher" /> |

* Start rviz:

|  |
| --- |
| <node name="rviz" pkg="rviz" type="rviz" args="-d $(arg rvizconfig)" required="true" /> |

* Full example launch file:

|  |
| --- |
| <launch> <arg name="model" default="$(find urdf\_tutorial)/urdf/01-myfirst.urdf"/>  <arg name="gui" default="true" />  <arg name="rvizconfig" default="$(find urdf\_tutorial)/rviz/urdf.rviz" />   <param name="robot\_description" command="$(find xacro)/xacro.py $(arg model)" />  <param name="use\_gui" value="$(arg gui)"/>   <node name="joint\_state\_publisher" pkg="joint\_state\_publisher" type="joint\_state\_publisher" />  <node name="robot\_state\_publisher" pkg="robot\_state\_publisher" type="state\_publisher" />  <node name="rviz" pkg="rviz" type="rviz" args="-d $(arg rvizconfig)" required="true" /> </launch> |

* To check xacro validity in terminal (cd to urdf file directory):

|  |
| --- |
| check\_urdf <(**xacro** samana.xml) |

* XML might be invalid because comment ending with triple minus --->

### Dependencies

|  |
| --- |
| # Check missing dependencies rosdep check --from-paths ~/Documents/SamanaAutonomousRobot/ROS/samana\_ws/src/ --ignore-src  # Check all dependencies (only installed?) rospack depends samana  # Simulate what dependencies will be installed rosdep install --simulate --from-paths ~/Documents/SamanaAutonomousRobot/ROS/samana\_ws/src/ --ignore-src  # Install missing dependencies rosdep install --from-paths ~/Documents/SamanaAutonomousRobot/ROS/samana\_ws/src/ --ignore-src A |

# STM32 in Hoverboard

* To stm32 flash:

|  |
| --- |
| cd to project folder make clean; make st-flash --reset write build/hover.bin 0x8000000 |

* STM32 config file if coreid is unexpected

|  |
| --- |
| /usr/share/opencd/scripts/target/stm32f1x.cfg |

* Ubuntu:
  + Write code in Arduino. Sketch -> Export compiled Binary
  + cd to binary

|  |
| --- |
| *st-flash --reset write <binary\_name>.bin 0x8000000* |

* /opt/STM32CubeProgrammer/bin/STM32CubeProgrammer
* To upload bootloder (On Window easy):
  + Dowload bootloader [generic\_boot20\_pc13.bin](https://github.com/rogerclarkmelbourne/STM32duino-bootloader/blob/master/binaries/generic_boot20_pc13.bin) (if led pin13):
    - <https://github.com/rogerclarkmelbourne/STM32duino-bootloader/tree/master/binaries>
  + Flash it in any way to the board (did with stlinkv2 and windows st-link util). Worked with both boot pins on 0
  + If DFU device not detected download and install driver (.bat both)
    - <https://codeload.github.com/rogerclarkmelbourne/Arduino_STM32/zip/master>

# Odometry

* Max forward speed 9 rps (rotations per second)
* Max one wheel speed turning 19 rps
* To measure covariances UMBMark test

# Hardware

* **IMPORTANT:** USB hub requires external power supply. Connecting the second laptop USB with double USB A works. Or connect power bank
* Raising the arm with 1kg object uses under 1000mA If pushed could raise around 2kg in the middle of the gripper. To fix it steal plate might be enough
* Weak point of lifter is it’s rack gear because under >1kg of weight it bends a lot
* Lowering without object uses <400mA
* Hoverboard max speeds:
  + No load:
    - Speed -4.4m/s - 4.6m/s
    - Angular speed -18rad/s - 15rad/s (-5.7rot/s - 4.8 rot/s)

# Mapping

* [range\_sensor\_layer](http://wiki.ros.org/action/fullsearch/range_sensor_layer?action=fullsearch&context=180&value=linkto%3A%22range_sensor_layer%22) for making costmap\_2d with sonar sensors

# Tips

* Use rosrun rosrun rqt\_top rqt\_top for profiling nodes for CPU and RAM usage
* roswtf for general debugging
* lwsm save <name> and lwsm restore <name> for saving and restoring layouts. <name> can be ommited. Might not open terminals on it’s own.

TODO BEFORE COMPETITION

# TODO

* Fully charge hoverboard
* Calibrate IMU heading
* Check laptop power settings: to be energy efficient, no automatic turn off!
* Hang small bag
* Disable manual arm motors control before autonomous mode
* Check if Arduinos are on right ports
* Check transmitter failsafe behaviour

# Needed tools

* Screw driver
* Hex tools (for grabber base plates)
* Soldering equipment
* All electronics (voltage regulator, motors, arduinos)
* Wires
* Electrical, masking, double sided tape
* Wrench 8mm?