**The Getting Started Doc for 2017-2018 IARC Team**

Welcome to IARC 2017-2018 Team. This document is to help all the subteams get started. Listed in each section will be some relevant topics for people to look into, read up on, google, download, or install. Please contact your sub-team leads with any questions.

***For everyone:***

* Register for our trello; this is how we manage tasks for a large team.
  + <https://trello.com/invite/b/YMlUbYBm/3cb8f93677d9bc7c009463f613c4b092/iarc-mission-7>
* Register for slack if you haven’t already; join the #iarc7 channel
  + pittras.slack.com
  + Use your **Pitt email** or it won’t let you in; talk to us if you have issues or aren’t a Pitt student
* Post your **gmail** in the #iarc7 slack so you can get added to the Google Drive folder
* Check out the presentation from the intro meeting: <https://docs.google.com/presentation/d/1TZScMtk0UKy74v2Au48rQyvzLR-sJVPl_xX5MNTyVko/edit?usp=sharing>
* Read our IARC 2016-2017 Postmortem to get an idea of where the team is at and what we need to do:
  + <http://pittras.org/projects/2017-iarc/2017/08/05/update-iarc-technical-postmortem.html>
  + Watch a video of us flying autonomously :D
    - <https://www.youtube.com/watch?time_continue=65&v=2w08D9MoOMk>
* Check out our GitHub wiki: <https://github.com/Pitt-RAS/iarc7_common>
* Fill out the when2meet to schedule the status meeting:
  + <https://www.when2meet.com/?6306615-g3jSS>
* **Come to the office hours to interact with leads and do some work! Message whoever you want to see there and make sure they can be there.**
  + **Sunday 4pm-9pm**
  + **Monday 5:30pm-9pm**
  + **Thursday 5:30pm-9pm**
* **Check out the section for your sub-team below**

***Setting up the software stack (simulator)***

* Control, perception, and planning team members will use the simulator to test new software
* We use ROS for all of our software. If you aren’t familiar with ROS, you can find a quick explanation of why it’s useful [here](http://www.ros.org/about-ros/) and [here](http://www.ros.org/is-ros-for-me/). To learn about how to use ROS in practice, including the concepts you’ll need to know for IARC, the [ROS tutorials](http://wiki.ros.org/ROS/Tutorials) are fantastic.
* **If you already have Ubuntu installed**
  + Follow the iarc7\_common simulator install instructions: <https://github.com/Pitt-RAS/iarc7_common/wiki/Installation>
* **If you have macOS or Windows**
  + Install VMWare Player (Windows only)
  + For those on macOS, install VirtualBox
  + Link to VMWare Player: <https://www.vmware.com/products/player/playerpro-evaluation.html>
  + Import this appliance: <https://drive.google.com/open?id=0B-WPRPndMAvmM2xndHJrQThwLVE>
* **Check out the wiki to learn about the software components:**
  + <https://github.com/Pitt-RAS/iarc7_common/wiki>
* **Learn to run test programs using this wiki page:**
  + <https://github.com/Pitt-RAS/iarc7_common/wiki/Launch-File-System>
* **Post your GitHub username in the #iarc7 channel so you can get added to our GitHub organization**
* **Check out these tutorials on Git:** 
  + [**http://learngitbranching.js.org**](http://learngitbranching.js.org)
  + [**http://product.hubspot.com/blog/git-and-github-tutorial-for-beginners**](http://product.hubspot.com/blog/git-and-github-tutorial-for-beginners)

***Electronics*:**

* Study current propulsion system (find out the motors and escs and such)
* Bug Levi about the thrust modeling rig since he didn’t write documentation on it
* New height find system (time of flight lidars) (Lightwave, lidar lite v3, v2, V53L0X
* look into using a lot of VL53L0X time of flight lidars
* Learn about **KiCad** (Hackaday has some fantastic articles on the library system)
* Check out allpcb.com
* thin-pot sensors (potentially used for roomba detection on side bumpers)
* pressure sensors for bottom feet
* Check out the KDE Direct product line (they’re an official sponsor)
* Google how quadcopters work
* Check out this link: <http://educypedia.karadimov.info/library/oneshots.pdf> to see how one-shot PWM to high/low circuits work (The implementation is similar, but different to the E-Kill)
* <https://bestdroneforthejob.com/drones-for-fun/racing-drone-buyers-guide-2/>

***Controls*:**

* Read articles on control theory linked here: <https://docs.google.com/spreadsheets/d/1Oa_iurNEQSbM15M0lVGJu3nz3lgQhV7fJAqV4DG5jFo/edit#gid=0>
* Definitely read about PID controllers, they’re essential.
* Learn about flight controllers, particularly ArduPilot (it’s also in that google sheet)
* Design sessions to design new version of low level motion that does not make the same mistakes as 2016-2017s low level motion
* Check out low level motion and FC\_comms (might be hairy, ask questions)
* Discuss low level motions current design (we should have a meeting)
* Learn about the Kalman filters, their characteristics affect controller design slightly (talk to perception team members)
* Learn about the controllers in ardupilot if you can
* Check out NVIDIA’s Github

https://github.com/NVIDIA-Jetson/redtail/wiki

***Perception*:**

* **State Estimation:**
  + Read about Kalman Filters
    - Decent intro here (not everything is entirely accurate)
      * <http://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/>
    - Detailed math on Wikipedia
      * <https://en.wikipedia.org/wiki/Kalman_filter>
  + Take a look at robot\_localization
    - <https://github.com/Pitt-RAS/robot_localization>
* **Obstacle Detection:**
  + Add depth cameras to drone in simulator
  + Research clustering algorithms like Meanshift (see resources link below)
  + Resources
    - <https://drive.google.com/open?id=0BzBxBZkfCGNXZmROSUJSeUtnV0U>
* **Roomba Detection:**
  + Read the wiki
    - <https://github.com/Pitt-RAS/iarc7_common/wiki/Roomba-Finding>
  + Read about Convolutional Neural Networks
  + Read about YOLO here
    - <https://pjreddie.com/darknet/yolo/>

***AI*:**

* Read over rules about mechanics of the competition
* Look at Ascend’s simulator (<https://github.com/AscendNTNU/ai-sim>), try getting it to work or developing our own
* Work on simple AI, like a state machine or similar
* See the resources here:
  + <https://drive.google.com/open?id=0BzBxBZkfCGNXdEtwYXNOQWFOLU0>

***Planning***:

* Read up on what planning is (see intro\_slides.pdf in planning folder)
* Check out the Wiki for High Level Motion here:
  + <https://github.com/Pitt-RAS/iarc7_common/wiki/Motion-Planner>
* Read papers on planning (see paper.pdf in planning folder)
* For papers and slides, please seem
  + <https://drive.google.com/drive/folders/0B9YArfzGNd6nMlYxUDBLaGhzNW8?usp=sharing>
* For search based planning, research A\* algorithm
  + <https://www.cs.cmu.edu/~./motionplanning/lecture/lecture.html>
* Read papers on software engineering found here:
  + <https://drive.google.com/drive/folders/0B-WPRPndMAvmN1FoM25SUEhqTnM>
* Read ‘No Silver Bullet’ also can be found at the above

***Mechanical*:**

* For system level design it’s important to understand basic UAV dynamics. Check out the articles linked here: <https://docs.google.com/spreadsheets/d/1uVyevgpGueDTfl4NgvzRhs0-5L06hMqr9MSmabFbJlQ/edit#gid=0>
* Check out the previous manufacturer list: <https://docs.google.com/spreadsheets/d/19mdzNLApzfIF_aS9JuJiOVhS2tgKmWobgjLyd-R6HDk/edit#gid=0>
* If you don’t know solidworks do the first couple solidworks tutorials:
  + The basement computers in 318 have solidworks on them and are perfect for this.
  + Use the slack to get ahold of a mech-e subteam lead if you need help
* Check out this guide on how to 3D Print smarter not harder: **We will add something soon plz google for now. We have a Makerbot 2X in the shop.**
* Read the rules to find out mechanical information: <http://www.aerialroboticscompetition.org/downloads/mission7rules_013017.pdf>