PART A

Fall 2017 Duckietown experience

Welcome to the Fall 2017 Duckietown experience.
UNIT A-1

The Fall 2017 Duckietown experience

This is the first time that a class is taught jointly across 3 continents!
There are 4 universities involved in the joint teaching for the term:

• ETH Zürich (ETHZ), with instructors Emilio Frazzoli, Andrea Censi, Jacopo Tani.
• University of Montreal (UdeM), with instructor Liam Paull.
• TTI-Chicago (TTIC), with instructor Matthew Walter.
• National Chiao Tung University (NCTU), with instructor Nick Wang.

This part of the Duckiebook describes all the information that is needed by the students of the four institutions.

At ETHZ, UdeM, TTIC, the class will be more-or-less synchronized. The materials are the same; there is some slight variation in the ordering.
Moreover, there will be some common groups for the projects.
The NCTU class is undergraduate level. Students will learn slightly simplified materials. They will not collaborate directly with the other classes.

1.1. The rules of Duckietown

The first rule of Duckietown

The first rule of Duckietown is: you don’t talk about Duckietown, using email.
Instead, we use a communication platform called Slack.
There is one exception: inquiries about “meta” level issues, such as course enrollment and other official bureaucratic issues can be communicated via email.

The second rule of Duckietown

The second rule of Duckietown is: be kind and respectful, and have fun.

The third rule of Duckietown

The third rule of Duckietown is: read the instructions carefully.
Do not blindly copy and paste.
Only run a command if you know what it does.
UNIT A-2
First Steps in Duckietown

2.1. Onboarding Procedure
Welcome aboard! We are so happy you are joining us at Duckietown!
This is your onboarding procedure. Please read all the steps and then complete all the
steps.
If you do not follow the steps in order, you will suffer from unnecessary confusion.

1) Github sign up
If you don’t already have a Github account, sign up now.
→ Github signup page
Please use your full name when it asks you. Ideally, the username should be something
like FirstLast or something that resembles your name.
When you sign up, use your university email. This allows to claim an educational dis-
count that will be useful later.

2) Questionnaire
For NCTU Students, complete this form:
NCTU Student Questionnaire
NCTU: Please complete by 10/30
For ETHZ, UdeM and TTIC fill in this Preliminary Student Questionnaire.
Zurich: Please fill in questionnaire by Tuesday, September 26, 15:00 (extended from
original deadline of 12:00).
Point of contact: if you have problems with this step, please contact Jacopo Tani
<tanij@ethz.ch>.

3) Accept invite to Github organization Duckietown
After we receive the questionnaire, we will invite you to the Duckietown organization.
You need to accept the invite; until you do, you are not part of the Duckietown organi-
zation and can’t access our repositories.
The invite should be waiting for you at this page.

4) Accept the invite to Slack
After we receive the questionnaire, we will invite you to Slack.
The primary mode of online confabulation between staff and students is Slack, a team
communication forum that allows the community to collaborate in making Ducki-
etown awesome.
(Emails are otherwise forbidden, unless they relate to a private, university-based ad-
ministrative concern.)
We will send you an invite to Slack. Check your inbox. If after 24 hours from sending the questionnaire you haven’t received the invite, contact HR representative Kirsten Bowser <akbowser@gmail.com>.  

**What is Slack?** More details about Slack are available [here](#). In particular, remember to disable email notifications.  

**Slack username.** When you accept your Slack invite, please identify yourself with first and last names followed by a “-” and your institution.  

*example* Andrea Censi - Zurich

**Slack picture.** Please add a picture (relatively professional, with duckie accessories encouraged).  

**Slack channels.** A brief synopsis of all the help-related Slack channels is here: [Unit A-10 - Slack Channels](#).  

Check out all the channels in Slack, and add yourself to those that pertain or interest you. Be sure to introduce yourself in the General channel.

### 2.2. (optional) Add Duckietown Engineering Linkedin profile

This is an optional step.

If you wish to connect with the Duckietown alumni network, on LinkedIn you can join the company “Duckietown Engineering”, with the title “Vehicle Autonomy Engineer in training”. Please keep updated your Linkedin profile with any promotions you might receive in the future.

### 2.3. Laptops

If you do not have access to a laptop that meets the following requirements, please post a note in the channel #help-laptops.  

You need a laptop with these specifications:

- Linux Ubuntu 16.04 installed natively (dual boot), not in a virtual machine. See [Subsection 2.3.1 - Can I use a virtual machine instead of dual booting?](#) below for a discussion of the virtual machine option.
- A WiFi interface that supports 5 GHz wireless networks. If you have a 2.4 GHz WiFi, you will not be able to comfortably stream images from the robot; moreover, you will need to adapt certain instructions.
- Minimum 50 GB of free disk space in addition to the OS. Ideally you have 200 GB+. This is for storing and processing logs.

There are no requirements of having a particularly good GPU, or a particularly good CPU. You will be developing code that runs on a Raspberry Pi. Any laptop bought in the last 3 years should be powerful enough. However, having a good CPU / lots of RAM makes it faster to run regression tests.

1) Can I use a virtual machine instead of dual booting?  

Running things in a virtual machine is possible, but not supported.
This means that while there is a way to make it work (in fact, Andrea develops in a VMWare virtual machine on OS X), we cannot guarantee that the instructions will work on a virtual machine, and, most importantly, the TAs will not help you debug those problems.

The issues that you will encounter are of two types.

- There are performance issues. For example, 3D acceleration might not work in the virtual machine.
- Most importantly, there are network configuration issues. These come up late in the class, when you start connecting the laptop to the Duckiebot. At that point, ROS makes certain assumptions about subnets, that might not be satisfied by your virtual machine configuration. At that point, you need to be relatively skilled to fix it.

So, the required skill here is not “being able to install Ubuntu on a virtual machine”, but rather “Being able to debug network problems involving multiple real/virtual networks and multiple real/virtual adapters”.

Here’s a quiz: do these commands look familiar to you?

```
$ route add default gw 192.168.1.254 eth0
$ iptables -A FORWARD -o eth1 -j ACCEPT
```

If so, then things will probably work ok for you. Otherwise, we strongly suggest that you use dual booting instead of a virtual machine.

### 2.4. Next steps for people in Zurich

1) Get acquainted with class journal and class logistics

At this point, you should be all set up, able to access our Github repositories, and, most important of all, able to ask for help on Slack. You can now get acquainted to the class journal, to know the next steps.

- [Unit A-11 - Zürich branch diary](#)

Also, in this page, we will collect the logistics information (lab times, etc.).

- [Unit A-3 - Logistics for Zürich branch](#)

2) Make sure you can edit the Duckuments

To receive your Duckiebox on Wednesday Sep 27, you need to prove to be able to edit the Duckuments successfully.

- See the instructions [in this section](#).

If you can’t come on Wednesday, please contact one of the TAs.

### 2.5. Next steps for people in Chicago
UNIT A-3
Logistics for Zürich branch

This section describes information specific to Zürich.

1) The local staff

These are the local TAs:
- Shiying Li (shili@student.ethz.ch)
- Ercan Selçuk (ercans@student.ethz.ch)
- Miguel de la Iglesia Valls (dmiguel@student.ethz.ch)
- Harshit Khurana (hkhurana@student.ethz.ch)
- Dzenan Lapandic (ldzenan@student.ethz.ch)
- Marco Erni (merni@ethz.ch)

Please contact them on Slack, rather than email.
Also feel free to contact the TAs in Montreal and Chicago.

3.1. HR

Feel free to contact Ms. Kirsten Bowser (akbowser@gmail.com) if you have problems regarding accounts, permissions, etc.

3.2. Website / class journal

During the term, we are not going to update the website.
Rather, all important information, such as deadlines, is in the class journal.
- Unit A-11 - Zürich branch diary

3.3. Duckiebox

The point of contact for Duckiebox distribution is Shiying Li.

3.4. Duckietown room access

The local Duckietown room is ML J 44.2.

comment
double check the room number -AC

3.5. Extra spaces

There will be extra lab space available.
Space-time coordinates TBD.
UNIT A-4
Logistics for Montréal branch

This unit contains all necessary info specific for students at Université de Montréal.

4.1. Website
This is the official course website. It contains links to the syllabus and description and other important info.

4.2. Class Schedule
The authoritative class schedule will be tracked in Unit A-12 - Montréal branch diary. This will contain all lecture material, homeworks, checkoffs, and labs.

4.3. Lab Access
The lab room for the class is 2333 in Pavillion André-Aisenstadt. The code for the door is XXX. Please do not distribute the code for the door, we are trying to limit access to this room as much as possible.

4.4. The Local Staff
The TA for the class is Florian Golemo. All communications with the course staff should happen through Slack.
The instructor is Prof. Liam Paull, whose office is 2347 Pavillion André-Aisenstadt.

4.5. Storing Your Robot
It is preferable that you keep your robot for the semester. However, if you do not have a secure location where you can store it, we can store it for you in Room XXX in Pavillion André-Aisenstadt. However, you will have to ask Prof. Liam Paull to access or store your robot there each time since we cannot give out access to this space to the students in the class.
UNIT A-5
Logistics for Chicago branch

Assigned to: Matt Walter
This section describes information specific to TTIC and UChicago students.

1) Website
The course website provides a copy of the syllabus, grading information, and details on learning objectives.

2) Class Schedule
Classes take place on Mondays and Wednesdays from 9am-11am in TTIC Room 530. In practice, each class will be divided into an initial lecture period (approximately one hour), followed by a lab session. The class schedule is maintained as part of the TTIC Class Diary, which includes details on lecture topics, links to slides, etc.

3) Course Grading
The following is taken from the course syllabus:
The class will assess your grasp of the material through a combination of problem sets, exams, and a final project. The contribution of each to your overall grade is as follows:
• 20%: Problem sets
• 10%: Checkoffs
• 20%: Participation
• 50%: Final project (includes report and presentation). The projects will be group-based, but we will assess the contribution of each student individually.
See the course syllabus for more information on how the participation and final project grades are determined.

4) Policy on Late Assignments and Collaboration
The following is taken from the course syllabus:
Late problem sets will be penalized 10% for each day that they are late. Those submitted more than three days beyond their due date will receive no credit.
Each student has a budget of three days that they can use to avoid late penalties. It is up to the student to decide when/how they use these days (i.e., all at once or individually). Students must identify whether and how many days they use when they submit an assignment.
It is not acceptable to use code or solutions from outside class (including those found online), unless the resources are specifically suggested as part of the problem set.
You are encouraged to collaborate through study groups and to discuss problem sets and the project in person and over Slack. However, you must acknowledge who you worked with on each problem set. You must write up and implement your own solu-
tions and are not allowed to duplicate efforts. The correct approach is to discuss solution strategies, credit your collaborator, and write your solutions individually. Solutions that are too similar will be penalized.

5) Lab Access

Duckietown labs will take place at TTIC in the robotics lab on the 4th floor. **Note:** TTIC and U. Chicago students in Matthew Walter’s research group use the lab as their exclusive research and office space. It also houses several robots and hardware to support them. Please respect the space when you use it: try not to distract lab members while they are working and please don’t touch the robots, sensors, or tools.

6) The Local LAs

Duckietown is a collaborative effort involving close interaction among students, TAs, mentors, and faculty across several institutions. The local learning assistants (LAs) at TTIC are:

- Andrea F. Daniele (afdaniele@ttic.edu)
- Falcon Dai (dai@ttic.edu)
- Jon Michaux (jMichaux@ttic.edu)
UNIT A-6
Logistics for NCTU branch

Assigned to: Nick and Eric (Nick’s student)
This section describes information specific to NCTU students.

1) Website
The Duckietown Taiwan Branch Website provides some details about Duckietown Branch in NCTU-Taiwan and results of previous class in NCTU.

2) Class Schedule
Classes take place on Thursday from 1:20pm~4:20om in NCTU Engineering Building 5 Room 635. Each class will be devided into two sessoins. In the first session, Professor Wang will give lessons on foundamental theory and inspire students to come up more creatitive but useful ideas on final projects. In the second sessoin, TAs will give pratical lab on how to use Duckietown platform as their project platform and use ROS as their middleware toward a fantastic work.
The class schedule is maintained as part of the NCTU Class Diary, which includes details on lecture topics, links to slides, etc.

3) Course Grading
The following is taken from the course syllabus:
This course aims at developing software projects usable in real-world, and focuses on “learning by doing,” “team work,” and “research/startup oriented.”. The contribution of each to your overall grade is as follows:
• Class Participation, In Class Quiz, Problem Sets (10%)
• Midterm Presentation (30%)
• Final Presentation (30%)
• Project Report and Demo Video (30%)
See the course syllabus for more information on course object and grading policy.

4) Policy on Late Assignments and Collaboration
The following is taken from the course syllabus:
Late problem sets will be penalized 10% for each day that they are late. Those submitted more than three days beyond their due date will receive no credit.
Each student has a budget of three days that they can use to avoid late penalties. It is up to the student to decide when/how they use these days (i.e., all at once or individually). Students must identify whether and how many days they use when they submit an assignment.
It is not acceptable to use code or solutions from outside class (including those found online), unless the resources are specifically suggested as part of the problem set. You are encouraged to collaborate through study groups and to discuss problem sets
and the project in person and over Slack. However, you must acknowledge who you worked with on each problem set. You must write up and implement your own solutions and are not allowed to duplicate efforts. The correct approach is to discuss solution strategies, credit your collaborator, and write your solutions individually. Solutions that are too similar will be penalized.

5) Lab Access

Duckietown labs will take place at NCTU in the same place with the lecture. **Note:** The course focus on “learning by doing” which means that the lab session of each class is especially important.

6) The Local LAs

Duckietown is a collaborative effort involving close interaction among students, TAs, mentors, and faculty across several institutions. The local learning assistants (LAs) at NCTU are:

- Yu-Chieh ‘Tony’ Hsiao (tonycar12002@gmail.com)
- Pin-Wei ‘David’ Chen (ccpwearth@gmail.com)
- Chen-Lung ‘Eric’ Lu (eric565648@gmail.com)
- Yung-Shan ‘Michael’ Su (michael1994822@gmail.com)
- Chen-Hao ‘Peter’ Hung (losttime1001@gmail.com)
- Hong-Ming ‘Peter’ Huang (peterx7803@gmail.com)
- Tzu-Kuan ‘Brian’ Chuang (fire594594594@gmail.com)
UNIT A-7

Git usage guide for Fall 2017

7.1. Differences
There are some differences among the branches. These will be marked using the following graphical notation.

?? | This is only for Zurich.
?? | This is only for Montreal.
?? | This is only for Chicago.
Taiwan | This is only for Taiwan.

7.2. Repositories
These are the repositories we use.

1) Software
The Software repository is the main repository that contains the software. The URL to clone is:

`git@github.com:duckietown/Software.git`

In the documentation, this is referred to as `DUCKIETOWN_ROOT`. During the first part of the class, you will only read from this repository.

2) Duckiefleet
The duckiefleet repository contains the data specific to this instance of the class. The URL to clone is:

`git@github.com:duckietown/duckiefleet.git`

In the documentation, the location of this repo is referred to as `DUCKIEFLEET_ROOT`. You will be asked to write to this repository, to update the robot DB and the people DB, and for doing exercises.

3) exercises-fall2017
For homework submissions, we will use the following URL:

`git@github.com:duckietown/exercises-fall2017.git`
As explained below, it is important that this repo is kept separate so that students can privately work on their exercises at schools where the homeworks are counted for grades.

4) Duckuments

The **Duckuments** repository is the one that contains this documentation.
The URL to clone is:

```bash
git@github.com:duckietown/duckuments.git
```

Everybody is encouraged to edit this documentation!
In particular, feel free to insert comments.

5) Lectures

The **lectures** repository contains the lecture slides.
The URL to clone is:

```bash
git@github.com:duckietown/lectures.git
```

Students are welcome to use this repository to get the slides, however, please note that this is a space full of drafts.

6) Exercises

The **exercises** repository contains the solution to exercises.
The URL to clone is:

```bash
git@github.com:duckietown/XX-exercises.git
```

Only TAs have read and write permissions to this repository.

### 7.3. Git policy for homeworks (TTIC/UDEM)

?? | This does not apply to Zurich.

Homeworks will require you to write and submit coding exercises. They will be submitted using git. Since we have a university plagiarism policy ([UdeM's](https://www.udem.ca/en), [TTIC/UChicago](https://www.ttic.uchicago.edu)) we have to protect students work before the deadline of the homeworks. For this reason we will follow these steps for homework submission:

1. Go [here](https://example.com) and file a request at the bottom “Request a Discount” then enter your institution email and other info.
2. Go to [exercises-fall2017](https://example.com)
3. Click “Fork” button in the top right
4. Choose your account if there are multiple options
5. Click on the Settings tab of your repository, not your account
6. Under “Collaborators and Teams” in the left, click the “X” in the right for the section for “Fall 2017 Vehicle Autonomy Engineers in training”. You will get a popup asking you to confirm. Confirm.
If you have not yet cloned the duckietown repo do it now:

```bash
$ git clone git@github.com:duckietown/exercises-fall2017.git
```

Now you need to point the remote of your exercises-fall2017 to your new local private repo. To do, from inside your already previously cloned exercises-fall2017 repo do:

```bash
$ git remote set-url origin git@github.com:GIT_USERNAME/exercises-fall2017.git
```

Let’s also add an upstream remote that points back to the original duckietown repo:

```bash
$ git remote add upstream git@github.com:duckietown/exercises-fall2017.git
```

If you type

```bash
$ git remote -v
```

You should now see:

```
origin   git@github.com: GIT_USERNAME/exercises-fall2017.git (fetch)
origin   git@github.com: GIT_USERNAME/exercises-fall2017.git (push)
upstream git@github.com:duckietown/exercises-fall2017.git (fetch)
upstream git@github.com:duckietown/exercises-fall2017.git (push)
```

Now the next time you push (without specifying a remote) you will push to your local private repo.

1) Duckiefleet file structure

You should put your homework files in folder at:

```
DUCKIEFLEET_HOMEWORK_ROOT/homeworks/ XX_homework_name / YOUR_ROBOT_NAME
```

Some homeworks might not require ROS, they should go in a subfolder called scripts. ROS homeworks should go in packages which are generated using the process described here: Unit J-6 - Minimal ROS node - pkg_name. For an example see DUCKIEFLEET_HOMEWORK_ROOT/homeworks/01_data_processing/shamrock.

**Note:** To make your ROS packages findable by ROS you should add a symlink from your DUCKIEFLEET_HOMEWORK_ROOT to duckietown/catkin_ws/src.
2) To submit your homework

When you are ready to submit your homework, you should do create a release and tag the Fall 2017 instructors/TAs group to let us know that your work is complete. This can be done through the command line or through the github web interface:

Command line:

```
$ git tag XX_homework_name -m "@duckietown/fall-2017-instructors-and-tas homework complete"
$ git push origin --tags
```

Through Github:

1. Click on the “Releases” tab.
2. Click “Create a new Release”.
3. Add a version (e.g. 1.0).
4. Release title put **XX_homework_name**.
5. In the text box put “@duckietown/fall-2017-instructors-and-tas homework complete”.
6. Click “Publish release”.

You may make as many releases as you like before the deadline.

3) Merging things back

Once all deadlines have passed for all institutions, we can merge all the homework. We will ask to create a “Pull Request” from your private repo.

1. In your private exercises-fall2017 repo, click the “New pull request button”.
2. Click “Create pull request” green button
3. The 4 drop down menus at the top should be left to right: (base fork: duckietown/exercises-fall2017, base: master, head fork: YOUR_GIT_NAME/exercises-fall2017, compare: YOUR_BRANCH)
4. Leave a comment if you like and click “Create pull request” green button below.
5. At some point a TA or instructor will either merge or leave you a comment.

4) For U de M students who have already submitted homework to the previous duckyfleet-2017 repo

These instructions assume that you are ok with losing the commit history from the first homework. If not, things get a little more complicated.

Fork and clone the new “homework” repository using the process above. Followed by:

```
$ git clone git@github.com:GIT_USERNAME/exercises-fall2017.git
```
Copy over your homework files from the duckiefleet-fall2017 repo into the exercises-fall2017 repo.

`git rm` your folder from duckiefleet-fall2017 and commit and push.
`git add` your folder to exercises-fall2017 and commit and push.

Clone the new duckiefleet repo

```bash
$ git clone git@github.com:duckietown/duckiefleet.git
```

Update the symlink you created in your duckietown repo

```bash
$ ln -sf EXERCISES_FALL2017/homeworks $DUCKIETOWN_ROOT/catkin_ws/src/name-of-the-symlink
```

7.4. Git policy for project development

Different than the homeworks, development for the projects will take place in the Software repo since plagiarism is not an issue here. The process is:

1. Create a branch from master
2. Develop code in that branch (note you may want to branch your branches. A good idea would be to have your own "master", e.g. "your_project-master" and then do pull requests/merges into that branch as things start to work)
3. At the end of the project submit a pull request to master to merge your code. It may or may not get merged depending on many factors.
UNIT A-8
Organization

The following roster shows the teaching staff.
Staff: To add yourself to the roster, or to change your picture, add a YAML file and a jpg file to the duckiefleet-fall2017 repository, in the people/staff directory.

8.1. The Activity Tracker

Link to Activity Tracker
The sheet called “Activity Tracker” describes specific tasks that you must do in a certain sequence. Tasks include things like “assemble your robot” or “sign up on Github”.
The difference between the Areas sheet and the Task sheet is that the Task sheet contains tasks that you have to do once; instead, the Areas sheet contains ongoing activities.
In this sheet, each task is a row, and each person is a column. There is one column for each person in the class, including instructors, TAs, mentors, and students.
You have two options:
• Only use the sheet as a reference;
• Use the sheet actively to track your progress. To do this, send a message to Kirsten with your gmail address, and add yourself.
Each task in the first column is linked to the documentation that describes how to perform the task.
The colored boxes have the following meaning:
• Grey: not ready. This means the task is not ready for you to start yet.
• Red: not started. The person has not started the task.
• Blue: in progress. The person is doing the task.
• Yellow: blocked. The person is blocked.
• Green: done. The person is done with the task.
• n/a: the task is not applicable to the person. (Certain tasks are staff-only.)

8.2. The Areas sheet
Please familiarize yourself with this spreadsheet and bookmark it in your browser.
The sheet called “Areas” describes the points of contact for each part of this experience. These are the people that can offer support. In particular, note that we list two points of contact: one for America, and one for Europe. Moreover, there is a link to a Slack channel, which is the place where to ask for help. (We’ll get you started on Slack in just a minute.)
UNIT A-9
Getting and giving help

9.1. **Who to ask for help**

1) **Primary points of contact**
   The organization chart ([Section 8.2 - The Areas sheet](#)) lists the primary contact for each area.

2) **Point of contacts for specific documents**
   Certain documents have specific points of contacts, listed at the top. These override the listing in the organization chart.

9.2. **How to ask for help**

The ways that we will support each other will depend on the type of situation. Here we will enumerate the different cases. Try to figure out which case is the most appropriate and act accordingly. These are ordered roughly in order of increasing severity.

1) **Case: You find a mistake in the documentation**
   **Action:** Please fix it.
   The goal for the instructions is that anybody is able to follow them. Last year, we managed to have two 15-year-old students reproduce the Duckiebot from instructions.
   → How to edit the documentation is explained in [Duckumentation documentation](#).
   
   Note that because we use Git, we can always keep track of changes, and there is no risk of causing damage.
   If you encounter typos, feel free to edit them directly.
   Feel free to add additional explanations.
   
   One thing that is very delicate is dealing with mistakes in the instructions.
   A few times the following happened: there is a sequence of commands cmd1;cmd2;cmd3 and cmd2 has a mistake, and cmd2b is the right one, so that the sequence of commands is cmd1;cmd2b;cmd3. In those situations we first just corrected the command cmd2.
   
   However, that created a problem: now half of the students had used cmd1;cmd2;cmd3 and half of the students had used cmd1;cmd2b;cmd3: the states had diverged. Now chaos might arise, because there is the possibility of “forks”.
   Therefore, if a mistaken instruction is found, rather than just fixing the mistake, please add an addendum at the end of the section.
   For example: “Note that instruction cmd2 is wrong; it should be cmd2b. To fix this, please enter then command cmd4“.
Later, when everybody has gone through the instructions, the mistake is fixed and the addendum is deleted.

2) Case: You find the instructions unclear and you need clarification

Action: Ask for clarification on the appropriate Slack channel. For a list of slack channels that could be helpful see Unit A-10 - Slack Channels. Once the ambiguity is clarified to your satisfaction, either you or the appropriate staff member should update the documentation if appropriate. For instructions on this see - Duckumentation documentation.

3) Case: You understand the instructions but you are blocked for some reason

Action: This is more serious than the previous. Open an issue on the duckiefleet-fall2017 github page. Once the issue is resolved, either you or the appropriate staff member should update the documentation if appropriate. For instructions on this see - Duckumentation documentation.

4) Case: You are having a technical issue related to building the documentation

Action: Open an issue on the duckuments github page and provide all necessary information to reproduce it.

5) Case: You have found a well-defined defect in the software.

Action: open an issue on the Software repository github page and provide all necessary information for reproducing the bug.
This page describes all of the helpful Slack channels and their purposes so that you can figure out where to get help.

10.1. Channels
You can also easily join the ones that you are interested in by clicking the links in this message.

<table>
<thead>
<tr>
<th>Channel</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>help-accounts</td>
<td>Info about necessary accounts, such as Slack, Github, etc.</td>
</tr>
<tr>
<td>help-assembly</td>
<td>Help putting your robot together</td>
</tr>
<tr>
<td>help-camera-calib</td>
<td>Help doing the intrinsic and extrinsic calibration of your camera</td>
</tr>
<tr>
<td>help-duckuments</td>
<td>Help compiling the online documentation</td>
</tr>
<tr>
<td>help-git</td>
<td>Help with git</td>
</tr>
<tr>
<td>help-infrastructure</td>
<td>Help with software infrastructure, such as Makefiles, unit tests,</td>
</tr>
<tr>
<td></td>
<td>continuous integration, etc.</td>
</tr>
<tr>
<td>help-laptops</td>
<td>Help getting your laptop setup with Ubuntu 16.04</td>
</tr>
<tr>
<td>help-parts</td>
<td>Help getting the parts for the robot or replacement parts if you broke</td>
</tr>
<tr>
<td></td>
<td>something</td>
</tr>
<tr>
<td>help-robot-setup</td>
<td>Help getting the robot setup to do basic things like be driven with a</td>
</tr>
<tr>
<td></td>
<td>joystick</td>
</tr>
<tr>
<td>help-ros</td>
<td>Help with the Robot Operating System (ROS)</td>
</tr>
<tr>
<td>help-wheel-calib</td>
<td>Help doing your odometry calibration</td>
</tr>
</tbody>
</table>

Note that we can link directly to the channels. (See list in the org sheet.) -AC
**UNIT A-11**

**Zürich branch diary**

11.1. Lectures and lab sessions

Lectures: Mon 13-15 HG F 26.5 Wed 10-12 HG E 22
Lab session: Fri 15-19 ML J 37.1
Duckielab: ML J 44.2

11.2. Wed Sep 20: Welcome to Duckietown!

This was an introduction meeting.

1) Material presented in class

These are the slides we showed:
- PDF
- Keynote (huge)

2) Feedback form

Please help us making the experience better by providing feedback (can be anonymous).

3) Pointers to reading materials

Read about Duckietown's history and watch the Duckumentary.
- Unit A-1 - The Duckietown project

Start learning about Git and Github.
- (unknown ref software_devel/git-reference)

Montreal, Chicago? What's happening?
- Unit A-1 - The Fall 2017 Duckietown experience

11.3. Monday Sep 25: Introduction to autonomy
1) Material presented in class

These are the slides we presented:
- Logistics: [Keynote](#), [PDF](#).
- Autonomous Vehicles: [Keynote](#), [PDF](#).
- Autonomous Mobility on Demand: [Keynote](#), [PDF](#).
- Plan for the next months: [Keynote](#), [PDF](#).

2) Feedback form

Please help us making the experience better by providing feedback (can be anonymous).

3) Questions and answers

**Q:** My Studies is not updated; I am still on the waiting list. What should I do?

**Answer:** Nothing. Don’t worry, if you have received the onboarding email, you are in the class, even if you still appear in the waiting list. We will figure this out with the department.

**Q:** What version of Linux do I need to install?

**Answer:** 16.04.* (16.04.03 is the latest at time of this writing)

**Q:** Do I need to install OpenCV, ROS, etc.?

**Answer:** Not necessary. We will provide instructions for those steps.

**Q:** My laptop is not ready. I’m having problems installing Linux on a partition.

**Answer:** Don’t worry, take a Duckie, and, take a breath. We have time to fix every issue. Start by asking for help in the #help-laptops channel in Slack. We will address the outstanding issues in the next classes.

**Q:** How much space do I need on my Linux partition?

**Answer:** At least 50 GB; 200 GB are recommended for easy processing of data (logs) later in the course. If you have less space (say ~100GB), it might be wise to acquire an external hard drive to use as storage.

**Q:** Are there going to be Linux training sessions?

**Answer:** Maybe. We didn’t plan for it, but it seems that there is a need. Subject to figuring out the logistics, we might organize an extra “lab” session or produce a support video.

11.4. Monday Sep 25, late at night: Onboarding instructions

At some late hour of the night, we sent out the onboarding instructions.

- [Section 2.1 - Onboarding Procedure](#)

Please complete the onboarding questionnaire by Tuesday, September 26, 15:00.

11.5. Wednesday Sep 27: Duckiebox distribution, and getting to know each other

Today we distribute the Duckieboxes and we name the robots. In other words, we per-
form the *Duckiebox ceremony*.

- getting to know each other;
- naming the robots;
- distribute the Duckieboxes.

**Note:** If you cannot make it to this class for the Duckiebox distribution, please inform the TA, to schedule a different time.

1) **Preparation, step 1: choose a name for your robot**

Before arriving to class, you must think of a name for your robot.

Here are the constraints:

- The name must work as a hostname. It needs to start with a letter, contains only letters and numbers, and no spaces or punctuation.
- It should be short, easy to type. (You’ll type it a lot.)
- It cannot be your own name.
- It cannot be a generic name like “robot”, “duckiebot”, “car”. It cannot contain brand names.

2) **Preparation, step 2: prepare a brief elevator pitch**

As members of the same community, it is important to get to know a little about each other, so to know who to rely on in times of need.

During the Duckiebox distribution ceremony, you will be asked to walk up to the board, write your name on it, and introduce yourself. Keep it very brief (30 seconds), and tell us:

- what is your professional background and expertise / favorite subject;
- what is the name of your robot;
- why did you choose to name your robot in that way.

You will then receive a Duckiebox from our senior staff, a simple gesture but of semipiternal glory, for which you have now become a member of the Duckietown community. This important moment will be remembered through a photograph. (If in the future you become a famous roboticist, we want to claim it’s all our merit.)

Finally, you will bring the Duckiebox to our junior staff, who will apply labels with your name and the name of the robot. They will also give you labels with your robot name for future application on your Duckiebot.

3) **Feedback form**

Please help us making the experience better by [providing feedback (can be anonymous)](#).

4) **Material presented in class**

- Duckiebot parts: [PowerPoint presentation](#), [PDF](#).

**11.6. Thursday Sep 28: Misc announcements**

We created the channel `#ethz-chitchat` for questions and other random things, so that we can leave this channel `#ethz-announcements` only for announcements.
We sent the final list to the Department; so hopefully in a couple of days the situation on MyStudies is regularized.

The “lab” time on Friday consists in an empty room for you to use as you wish, for example to assemble the robots together. In particular, it’s on the same floor of the Duckietown room and the IDSC lab.

The instructions for assembling the Duckiebots are [here](#opmanual_duckiebot/rc-cam-launched-remotely). Note that you don’t have to do the parts that we did for you: buying the parts, soldering the boards, and reproducing the image.

Expected progress: We are happy if we see everybody reaching up to (unknown ref opman-

A couple of announcements:

1. We created #ethz-chitchat for questions and other random things, so that we can leave this channel #ethz-announcements only for announcements.
2. MyStudies should be updated with everybody’s names.
3. The “lab” time tomorrow consists in an empty room for you to use as you wish, for example to assemble the robots together. In particular, it’s on the same floor of the Duckietown room and the IDSC lab.
4. The instructions for assembling the Duckiebots are [here](#opmanual_duckiebot/rc-cam-launched-remotely). Note that we did for you step I-2 (buying the parts) and I-3 (soldering the boards); and I-6 is optional.
5. We are happy if we see everybody reaching I-13 by the Monday after next. I encourage you to start sooner than later.
6. I see only 30 people in this channel instead of 42. Please tell your friends that now all the action is on Slack.

It looks like that the current documentation is misleading in a couple of points. This is partially due to the fact that there is some divergence between Chicago, Montreal and Zurich regarding (1) the parts given out and (2) the setup environment (which networks are available). We did the simple changes (e.g. removing the infamous part 6), but we need some more time to review the other issues. At this point, the best course
of action is that you enjoy your weekend without working on Duckietown, while we spend the weekend fixing the documentation.

11.9. Oct 02 (Mon): Networking, logical/physical architectures

1) Materials presented in class

a - Logistics and other information: Keynote, PDF.
b - Networking Keynote, PDF.
c - System architecture basics Keynote, PDF.

2) Feedback form

Please help us making the experience better by providing feedback (can be anonymous)

11.10. Oct 04 (Wed): Modeling

1) Materials presented in class

- Modeling of a differential drive vehicle: PowerPoint presentation, PDF.

2) Feedback form

Please help us making the experience better by providing feedback (can be anonymous)

11.11. Oct 09 (Mon): Autonomy architectures and version control

1) Materials presented in class

ZÜRICH BRANCH OPENS
Duckies rejoice, hopeful of swift construction of AMOD service.

Who are the men without the ducks?
Speculations run wild regarding the identity and motives of three men who were in the picture without a duckie on their head.

Instruction confusion leads to assembling halt
Following the sudden realization that Montreal, Chicago, and Zurich have 3 different robot configurations, assembly and setup has been stopped while the instructions are being updated. Officials are hopeful that there will be a quick convergence of all branches to same configuration.

Rumors of Chicago and Montréal branches opening
No photographic evidence to be found.

Duckietown construction started
Construction suddenly halted when crew realized they were upside down.

Duckie boxes distributed
Commencement day boxes are observed to be similar to stacks of Swiss cheese. Boxes planned in future upgrade.

Popular vote dictates networking lecture
Anonymous poll said to bias the results. Anonymous poll planned in the future.

Duckietown critiqued over use of non-free software
Free software advocate Richard Stallman criticizes Duckietown over the choice of GitHub due to the website's use of non-free software.

Figure 11.2

1) **Materials presented in class**
   a - Computer Vision Basics: PDF, PowerPoint presentation.
   b - Odometry Calibration: PDF, PowerPoint presentation.

11.13. Oct 13 (Fri): **new series of tasks out**

1) **Taking a video of the joystick control**
   Please take a video of the robot as it drives with joystick control, as described in Section 16.7 - Upload your video and upload it according to the instructions. Example of a great video, but with a nonconforming Duckiebot.

2) **Camera calibration**
   Go forth and calibrate the camera! And get help in #help-camera-calib.

   3) **Wheel calibration**
   This is not ready yet! will be ready in a day or so.

4) **Taking a log check off**
   Follow the instructions here to learn how to take a log.

5) **Data processing exercises**
   See the list of exercises here.
   Get help in #ex-data-processing.

| Table 11.2. CURRENT DEADLINES FOR ZURICH |
|-----------------------------------------|-----------------------------|
| Robot assembly                          | overdue                    |
| Robot/laptop configuration              | overdue                    |
| Subsection 11.13.1 - Taking a video of the joystick control | Monday Oct 16 |
| Subsection 11.13.2 - Camera calibration | Friday Oct 20              |
| Subsection 11.13.3 - Wheel calibration  | not ready yet              |
| Subsection 11.13.4 - Taking a log check off | Wed Oct 18               |
| Subsection 11.13.5 - Data processing exercises | Monday Oct 23             |


1) **Materials presented in class**
   a - Line detection: Keynote, PDF.
b - Logistics: [Keynote, PDF].

11.15. Oct 18 (Wed): Feature extraction

1) Materials presented in class
   • Feature extraction: [Keynote, PDF].

11.16. Oct 20 (Fri): Lab session

1) Materials presented in class
   • ROS Main concepts: [PowerPoint presentation, PDF].

11.17. Oct 23 (Mon) Filtering I

1) Materials presented in class
   • Lectures on filtering (Filtering I): [PowerPoint presentation, PDF].

11.18. Oct 25 (Wed) Filtering II

a - Lectures (Particle Filter) [PowerPoint presentation, PDF].
b - Lectures (Lane Filter) [PowerPoint presentation, PDF].


a - Lectures (Control Systems Module I) [PowerPoint presentation, PDF].
b - Lectures (Control Systems Module II) [PowerPoint presentation, PDF].

Points to be noted - Running what-the-duck on laptop and Duckiebot is mandatory. It helps save time in debugging errors and also is a standard way to ask for help from the staff. Keep repeating it periodically so as to keep the data up-to-date - For the people lacking calibrated wheels, this serves as a reminder to calibrate the wheels and keep their duckiebot up-to-date - It is advised to fill the lecture feedback form (Feedback form), so as to increase the effectiveness of the lectures - Always check the consistency of the camera calibration checkerboard before camera calibration (one has to check for the correct square size and correct distance between world and checkerboard reference).

11.20. Nov 6 (Mon) Project Pitches

Lecture Project Pitches [PDF].


Lecture Motion Planning [PDF].
A few references for planning of Andrea Censi:

- The Book on planning is the one by Lavalle, available for free here: [http://planning.cs.uiuc.edu/](http://planning.cs.uiuc.edu/).
- The mentioned movie scene Donnie Darko: [https://www.youtube.com/watch?v=rPeGaos7DB4](https://www.youtube.com/watch?v=rPeGaos7DB4).

### 11.22. Nov 13 (Mon) Project Team Assignments

- First Lecture: Project Team Assignments [PDF](http://example.com).
- Second Lecture: First meeting of the Controllers group –> Filling out the Preliminary Design Document

### 11.23. Nov 15 (Wed) Putting things together

- First Lecture: Putting things together [PDF](http://example.com).
- Second Lecture: Second meeting of the Controllers group –> Filling out the Preliminary Design Document

### 11.24. Nov 20 (Mon) Testing Autonomous Vehicles

- First Lecture: Testing Autonomous Vehicles [PDF](http://example.com).

### 11.25. Nov 22 (Wed) Fleet Control

- Lecture: Fleet Control in Autonomous Mobility on Demand [PDF](http://example.com).


- First Lecture: The intermediate Design report is introduced [here](http://example.com).

> I will ignore this because it is an external link.

I do not know what is indicated by the link '#class_2017_projects/template-int-report'.

It is due on Monday 4th of December.
- Second Lecture was left for project discussion and interaction.
11.27. Nov 29 (Wed) Fleet Control

- First Lecture: Claudio finished Fleet Control in Autonomous Mobility on Demand PDF.
- Second Lecture Julian Presented the state of the art in data driven vs Model driven robotics. PDF
UNIT A-12
Montréal branch diary

12.1. Wed Sept 6
Class (11:30)
Slides:
• Duckietown History Future (keynote) (pdf)
• Duckietown Intro (keynote) (pdf)
• Autonomy Overview (keynote) (pdf)
• Autonomous Vehicles (keynote) (pdf)
Book materials:
• Unit A-1 - The Duckietown project
• Unit A-1 - Autonomous Vehicles
• Unit A-2 - Autonomy overview

12.2. Friday Sept 8
Acceptance emails sent

12.3. Sun Sept 10
• Onboarding email sent to accepted students

12.4. Mon Sept 11
Class (10:30)
Note: This class we are meeting in rm. 2333 Pavillion André Aisenstadt.
• Logistics
• The Duckiebook
• Slack (Unit A-10 - Slack Channels)
• Git repos (Unit A-7 - Git usage guide for Fall 2017)
• How to get and give help (Unit A-9 - Getting and giving help)
• Grading scheme
• Student/Staff Introductions
• Duckiebox distribution.
• Go through the Duckiebox parts
• Unit A-16 - Checkoff: Assembly and Configuration initiated.
Deadline: Mon Sept. 25

12.5. Wed Sept 13
Class canceled. Continue working on Unit A-16 - Checkoff: Assembly and Configuration.

12.6. Mon Sept 18

Class (10:30 - 11:30)
- General discussion (how are things going? Sorry I was away last week.. anyone need anything?)
- Intro to robotics - Modern robotic systems
- The robot as a system - System architectures
- Decomposing the robotics systems into smaller pieces - autonomy architectures
- Agreeing on the language that the different pieces “talk” - representations
- Background on basic probability theory?

Slides:
- Modern Robotic Systems (keynote) (pdf)
- System Architecture (keynote) (pdf)
- Autonomy Architectures (keynote) (pdf)
- Representations for Robotics (keynote) (pdf)

Book Materials:
- Modern Robotic Systems
- System Architecture Basics
- Autonomy Architectures
- Representations (unknown ref preliminaries/representations)

Warning: I will ignore this because it is an external link.

> I do not know what is indicated by the link '#preliminaries/representations'.

Location not known more precisely.

Created by function check_if_any_href_is_invalid in module mcdp_docs.check_missing_links.

- Probability Basics

Lab (11:30 - 12:30)
- Liam will be in 2333 setting up network and building Duckietown.

12.7. Wed Sept 20

Class (11:30 - 12:30)
- Robotics middlewares - what are they and basic concepts
- Introduction to the Robot Operating System (ROS)

Slides:
montréal branch diary

- Software Architectures (keynote) (pdf)
- Introduction to ROS (keynote) (pdf)

Book Materials:
- Introduction to ROS
- Middlewares (unknown ref learning_materieals/middleware)

previous warning next (5 of 24) index

warning

I will ignore this because it is an external link.

> I do not know what is indicated by the link '#learning_materieals/middleware'.

Location not known more precisely.
Created by function check_if_any_href_is_invalid in module mcdp_docs.check_missing_links.

Lab (12:30 - 1:30)

Homeworks and Checkoffs:

12.8. Mon Sept 25

Class (10:30-11:30)
- Microelectronics (pptx) (pdf)
- Modern Signal Processing (keynote) (pdf)
- Intro to Networking (keynote) (pdf)

Book Materials (still rough drafts, will be completed soon):
Lab (11:30 - 12:30)
- Liam will be in 2333
- Any final help needed for Unit A-16 - Checkoff: Assembly and Configuration
- Finalize Duckietown map setup

Homeworks and Checkoffs:
- Unit A-16 - Checkoff: Assembly and Configuration due by 11pm.
- New checkoff: “Taking a log” will be linked later today (due Monday Oct 2)
- New homework: “Data processing” will be linked later today (due Monday Oct 2)

12.9. Wed Sept 27

Class (11:30 - 12:30) in Z-305
- USB drive distribution
- New checkoff announced: Unit A-17 - Checkoff: Take a Log
- New homework announced: Unit A-19 - Homework: Data Processing (UdeM)
Let’s review the git policy for homeworks: Section 7.3 - Git policy for homeworks (TTIC/UDEM)

Announcement regarding activity spreadsheet which is now embedded here: Section 8.1 - The Activity Tracker. Feel free to just look or send Kirsten your gmail on Slack and she will give you access to it.

Slides:
- Duckiebot Modeling (pptx) (pdf)

Book Material:
- Unit B-5 - Coordinate systems
- Unit B-6 - Reference frames
- Unit E-1 - Duckiebot modeling

12.10. Mon Oct 2

Class (10:30 - 11:30) in Z-210

Slides:
- Computer vision basics: (keynote) (pdf)

Book Material:
- Unit C-1 - Computer vision basics
- Unit C-2 - Camera geometry
- Unit C-3 - Camera calibration
- Unit C-4 - Image filtering

Lab (11:30 - 12:30) in AA 2333

12.11. Wed Oct 4

Class (11:30 - 12:30) in Z-305

Slides:
- Computer Vision - Lane and Line Detection (keynote) (pdf)

Book Material:
- Unit C-4 - Image filtering
- Unit C-7 - Line Detection

Lab (12:30 - 1:30) in AA 2333

| Note: Checkoff and Homework due at 11pm

12.12. Mon Oct 9

Holiday no class!


Class 11:30 in Z-305

- Computer Vision - Feature descriptors (keynote) (pdf)
Book Material:
Lab 12:30 - 1:30 in AA2333

New Checkoff Initiated: Unit A-18 - Checkoff: Robot Calibration Deadline is Monday Oct 23. Deliverables are: - Screenshot of your robot passing the kinematic odometry test - PR to duckiefleet repo with your 3 robot calibrations (kinematics, camera intrinsics, camera extrinsics)

12.15. Monday Oct 16
Class 10:30-11:30 Z-205
- Reminder about checkoff.
- Intro to filtering (pptx) (pdf)

12.16. Wednesday Oct. 18
Class 12:30-1:30 Z-310
Guest Lecture from Prof. James Forbes from McGill on Extended Kalman filter slides: (pdf)

12.17. Friday Oct. 20
Homework Unit A-22 - Homework: Augmented Reality announced. Deadline is Friday Oct. 27 at 11pm

12.18. Monday Oct. 23
No class (Reading Week)

12.19. Wednesday Oct. 25
No class (Reading Week)

12.20. Monday Oct. 30
Class 10:30-11:30 Z-2015
Start of Introduction to SLAM
Slides: (pdf) (keynote)

12.21. Wednesday Nov. 1
Class 11:30-12:30
End of Introduction to SLAM
12.22. Monday Nov. 6
Class 10:30-12:30
Project Pitches. [Link to slides]

12.23. Wednesday Nov. 8
Class 11:30-12:30
- Motion Planning. [pdf], [pptx]
  Checkoff Navigation [unknown ref fall2017_info/checkoff_navigation]

  I will ignore this because it is an external link.

  > I do not know what is indicated by the link '
  #fall2017_info/checkoff_navigation'.

  Location not known more precisely.

  Created by function check_if_any_href_is_invalid in module
  mcdp_docs.check_missing_links.

  initiated. Deadline Nov 15 @ 11pm.

12.24. Thursday Nov. 9
- Filtering Homework initiated. Deadline Nov 17 @ 11pm.
- Project groups announced

12.25. Monday Nov. 13
Class 10:30-11:30
Control.
Module 1: [pptx]
Module 2: [pptx]
Module 3: [pptx]

12.26. Wednesday Nov. 15
Class 11:30-11:30
David Vazquez guest lecture
UNIT A-13
Chicago branch diary

Classes take place on Mondays and Wednesdays from 9am-11am in TTIC Room 530.

13.1. Checkoffs:
The following is a list of the current checkoffs, along with the date and time when they are due. Most submissions involve uploading videos to via Dropbox File Request using the link provided in the individual checkoff.

- **Friday October 6, 5pm CT**: Unit A-16 - Checkoff: Assembly and Configuration
- **Wednesday October 18, 11:59pm CT**: Unit A-17 - Checkoff: Take a Log
- **Sunday October 22, 11:59pm CT**: Unit A-18 - Checkoff: Robot Calibration
- **Wednesday November 15, 11:59pm CT**: Unit A-23 - Checkoff: Navigation

13.2. Problem Sets:
The following is a list of the current problem sets, along with the date and time when they are due.

- **Section 7.3 - Git policy for homeworks (TTIC/UDEM)**

for instructions on how the homework should be submitted.

---

**Note:** Please keep track of how much time you spend on each problem set. We will ask you for this estimate along with other feedback at the end of each problem set.

- **Friday October 13, 11:59pm CT**: Unit A-20 - Homework: Data Processing (TTIC)
- **Friday October 27, 11:59pm CT**: Unit A-22 - Homework: Augmented Reality
- **Friday November 17, 11:59pm CT**: Unit A-24 - Homework: Lane Filtering

13.3. Monday September 25: Introduction to Duckietown

1) Lecture Content
   - Duckietown Course Intro ([Keynote, PDF](#))

2) Reading Material
   - Unit A-1 - The Duckietown project
   - Unit A-2 - Autonomy overview

3) Feedback Form
   Please help us making the experience better by [providing feedback (can be anonymous)](#)
13.4. Tuesday September 26: Onboarding
Tonight, we sent out the onboarding instructions.

- Section 2.1 - Onboarding Procedure

Please complete the onboarding questionnaire by Thursday, September 27, 5:00pm CT

13.5. Wednesday, September 27: Duckiebox Ceremony
Welcome to Duckietown! This lecture constitutes the Duckiebox ceremony during which we will distribute your Duckieboxes and ask you to name your yet-to-be built Duckiebots! We will also discuss logistics related to the course, but we admit that that isn’t as exciting.

1) Lecture Content
- The Duckiebook
- Slack (Unit A-10 - Slack Channels)
- Git repos (Unit A-7 - Git usage guide for Fall 2017)
- How to get and give help (Unit A-9 - Getting and giving help)
- Unit A-16 - Checkoff: Assembly and Configuration initiated
- Getting to know one another
- Naming your robots
- Distribute the Duckieboxes!
As you name your Duckiebots, please consider the following constraints:
- The name must work as a hostname. It needs to start with a letter, contains only letters and numbers, and no spaces or punctuation.
- It should be short, easy to type. (You’ll type it a lot.)
- It cannot be your own name.
- It cannot be a generic name like “robot”, “duckiebot”, “car”. It cannot contain brand names.

2) Feedback Form
Please help us making the experience better by providing feedback (can be anonymous)

13.6. Monday, October 2: Modern Robotic Systems
1) Lecture Content
- Logistics (Keynote, PDF)
- Autonomous Vehicles (Keynote, PDF)
- Modern Robotic Systems (Keynote, PDF)
- System Architecture Basics (Keynote, PDF)

2) Reading Material
• **Unit A-1 - Autonomous Vehicles**
• **Unit A-2 - Autonomy overview**
• **Unit B-1 - Modern Robotic Systems**
• **Unit B-6 - System architectures basics**

3) Assignments

• **Unit A-16 - Checkoff: Assembly and Configuration** is due by Friday 5pm CT. Note: The page provides the URL where you should upload your video(s).

4) Feedback Form

Please help us making the experience better by providing feedback (can be anonymous)

13.7. Wednesday, October 4: Modern Robotic Systems (Continued)

Note: Today’s lecture will take place in Room 501 due to the TTIC Board Meeting

1) Lecture Content

• Logistics (**Keynote, PDF**)
• Representations (**Keynote, PDF**)
• Software Architectures (**Keynote, PDF**)
• Networking (**Keynote, PDF**)

2) Reading Material

• (unknown ref preliminaries/representations)

```
previous warning next (7 of 24) index

I will ignore this because it is an external link.

> I do not know what is indicated by the link '#preliminaries/representations'.

Location not known more precisely.
Created by function check_if_any_href_is_invalid in module mcdp_docs.check_missing_links.
```

• (unknown ref learning_materials/networking)

```
previous warning next (8 of 24) index

I will ignore this because it is an external link.

> I do not know what is indicated by the link '#learning_materials/networking'.

Location not known more precisely.
Created by function check_if_any_href_is_invalid in module mcdp_docs.check_missing_links.
```
3) Feedback Form

Please help us making the experience better by providing feedback (can be anonymous)

13.8. Monday, October 9: Modeling

1) Preparation

Important: Before starting these tutorials, make sure that you completed the following:

- (unknown ref opmanual_duckiebot/setup-laptop)

Before coming to class, please read through the following tutorials:

- [https://tinyurl.com/ROS101-Intro](https://tinyurl.com/ROS101-Intro)
- [http://wiki.ros.org/ROS/Tutorials#Beginner_Level](http://wiki.ros.org/ROS/Tutorials#Beginner_Level) Complete all the tutorials in Section 1.1 Beginner Level. There is no need to install ROS as the tutorial instructs, because you already installed as part of the setup process (unknown ref opmanual_duckiebot/setup-laptop)

. Sections 11 and 14 of this tutorial will ask you to implement simple ROS nodes using C++. You can skip them since Sections 12 and 15 are basically the same but using Python.

2) Lecture Content

- Modeling ([Powerpoint, PDF](#))
3) Reading Material
- Unit E-1 - Duckiebot modeling

4) Feedback Form
Please help us making the experience better by providing feedback (can be anonymous)

13.9. Wednesday, October 11: Introduction to Computer Vision

1) Lecture Content
- Introduction to Computer Vision (Keynote, PDF)
- Camera Models (Keynote, PDF)

2) Reading Material
- Unit C-1 - Computer vision basics
- Unit C-2 - Camera geometry
- Richard Szeliski, Computer Vision: Algorithms and Applications, Chapters 1 and 2 (available online)
- David A. Forsyth and Jean Ponce, Computer Vision: A Modern Approach, Chapters 1 and 2

3) Feedback Form
Please help us making the experience better by providing feedback (can be anonymous)

13.10. Monday, October 16: Camera Calibration and Image Filtering

| Note: The second checkoff is due by 11:59pm CT Wednesday. |

1) Lecture Content
- Logistics (Keynote, PDF)
- Calibration (Keynote, PDF)
- Image Filtering (Keynote, PDF)

2) Reading Material
- Unit C-1 - Computer vision basics
(unknown ref opmanual_duckiebot/camera-calibration)

I will ignore this because it is an external link.

> I do not know what is indicated by the link '#opmanual_duckiebot/camera-calibration'.
Location not known more precisely.
Created by function check_if_any_href_is_invalid in module mcdp_docs.check_missing_links.

- **Unit C-4 - Image filtering**
- Richard Szeliski, *Computer Vision: Algorithms and Applications*, Chapters 3 and 6 (available online)
- David A. Forsyth and Jean Ponce, *Computer Vision: A Modern Approach*, Chapters 5.3 and 7

3) Feedback Form

Please help us making the experience better by providing feedback (can be anonymous)

**13.11. Wednesday, October 18: Edge Detection and Lane Detection**

**Note:** The third checkoff is due by 11:59pm CT Sunday. Note that camera calibration is necessary for the next problem set, which will be posted soon.

1) Lecture Content

- Logistics ([Keynote, PDF](#))
- Image Filtering (Review) ([Keynote, PDF](#))
- Image Gradients ([Keynote, PDF](#))
- Edge Detection ([Keynote, PDF](#))
- Line Detection ([Keynote, PDF](#))
- Lane Detection ([Keynote, PDF](#))

2) Reading Material

- **Unit C-7 - Line Detection**
- David A. Forsyth and Jean Ponce, *Computer Vision: A Modern Approach*, Chapters 7 and 8

3) Feedback Form

Please help us making the experience better by providing feedback (can be anonymous)

**13.12. Monday, October 23: Feature Detection and Place Recognition**

1) Lecture Content

- Logistics ([Keynote, PDF](#))
- Robust Fitting ([Keynote, PDF](#))
- Feature Detection ([Keynote, PDF](#))
- Place Recognition ([Keynote, PDF](#))
2) Reading Material
• Richard Szeliski, *Computer Vision: Algorithms and Applications*, Chapters 4 and 6.1.3 (available online)

3) Feedback Form
Please help us making the experience better by providing feedback (can be anonymous)

13.13. Wednesday, October 25: Filtering I

1) Lecture Content
• Place Recognition (continued) ([Keynote](#), [PDF](#))
• Introduction to Filtering ([Powerpoint](#), [PDF](#))

2) Reading Material
• Sebastian Thrun, Wolfram Burgard, and Dieter Fox, *Probabilistic Robotics*, Chapters 1 and 2

3) Feedback Form
Please help us making the experience better by providing feedback (can be anonymous)


1) Lecture Content
• Nonparametric Filtering ([Powerpoint](#), [PDF](#))

2) Reading Material
• Sebastian Thrun, Wolfram Burgard, and Dieter Fox, *Probabilistic Robotics*, Chapter 4

3) Feedback Form
Please help us making the experience better by providing feedback (can be anonymous)

13.15. Wednesday, November 1: Introduction to SLAM

1) Lecture Content
• SLAM Intro ([Keynote](#), [PDF](#))

2) Reading Material
• Sebastian Thrun, Wolfram Burgard, and Dieter Fox, *Probabilistic Robotics*, Chapters 9, 10, and 13
3) Feedback Form

Please help us making the experience better by providing feedback (can be anonymous)


1) Lecture Content
   - Planning Intro (Keynote, PDF)
   - Project Pitches (Google Slides)

2) Reading Material
   - Steven M. LaValle, Planning Algorithms, Chapters 3, 4, and 6. Available online

3) Feedback Form

Please help us making the experience better by providing feedback (can be anonymous)

13.17. Wednesday, November 8: Introduction to Planning (Continued)

1) Lecture Content
   - Planning Intro (Keynote, PDF)

2) Reading Material
   - Steven M. LaValle, Planning Algorithms, Chapter 5. Available online

3) Feedback Form

Please help us making the experience better by providing feedback (can be anonymous)

13.18. Monday, November 13: Introduction to Control

1) Lecture Content
   - Control Intro (Powerpoint, PDF)

2) Feedback Form

Please help us making the experience better by providing feedback (can be anonymous)

13.19. Wednesday, November 15: Introduction to Control (Continued)

1) Lecture Content
   - Controls (Powerpoint, PDF)
   - Controls for Duckietown (Powerpoint, PDF)
2) Feedback Form
Please help us making the experience better by providing feedback (can be anonymous)


1) Lecture Content
• Testing for Autonomous Vehicles (Keynote, PDF)

2) Feedback Form
Please help us making the experience better by providing feedback (can be anonymous)
Classes take place on Thursday from 1:20pm-4:20pm in NCTU Engineering Building 5 Room 635.

14.1. Checkoffs:
The following is a material of the current preview lecture, along with the date and time when they are due. Please make sure that you preview the materials every week before the class for insuring a good learning quality.

- **Thursday October 19, 1:20pm**: [Week5 Material]
- **Thursday October 26, 1:20pm**: [Week6 Material]

14.2. Course Material:
The following is a list of the course material. Please preview the course every week and better if you try the lab yourself.

- **Course Material**

- **Thursday October 12, 1:20pm**: [Week5 Material]

14.3. Thursday September 14: Introduction to Duckietown and Creative Software Project

1) Lecture Content
- Duckietown Course Intro ([Week1 Material])

14.4. Thursday September 21: Project Ideas
How do you choose a good project idea? How about “writing” a good project idea?

- **Week2 Material**

1) Lecture Content
- **Week2 Lecture**

2) Weekly Lab
This is the first lab of the semester. We are going to teach you “git” which is essential
when becoming a professional programmer and cooperating with professional team.

- **Week2 Lab**

**14.5. Thursday September 28: Robotics System**

What is a robotics system? We'll introduce the concept of robotics system and some well known software architecture and middleware. Also, robotic operation system will be introduced in this class.

See” **Week3 Material**

1) Lecture Content

- **Week3 Lecture**

2) Weekly Lab

This week we'll continue on the topic of git. Also, a new exciting chapter has been opened. We are going to prepare our own Duckiebot! Come and join us!.

- **Week3 Lab**

**14.6. Thursday October 5: OpenCV, Python and Jupyter Notebook**

Today we're going deeply into Duckiebot's “mind”. What is the algorithm behind lane following? What is the secret that the duckies are so smart to drive? Here comes the answer.

- **Week4 Material**

1) Lecture Content

- **Week4 Lecture**

2) Weekly Lab

Jupyter notebook is a very useful and convenient tool while dealing with python language. We will teach you how to use it. A part of lane following algorithm will be taught this week which is about line detector.

- **Week4 Lab**

**14.7. Thursday October 19: Camera and Wheel Calibration**

This week we are going to do the camera and wheel calibration. We will teach the student the theory of camera calibration, including extrinsic and intrinsic.

- **Week6 Material**

1) Lecture Content
2) Weekly Lab

We will let them control the duckiebots by joystick to finish the wheel calibration. Also, we will give them chessboard for camera calibration.

- Week6 Lab

14.8. (Template for every lecture) Date: Topic

What is this lecture all about?

1) Preparation

Things that the students should do before class.

2) Lecture Content

Link to PDF and Keynote/Powerpoint materials.

3) Feedback Form

Please help us making the experience better by providing feedback (can be anonymous)

4) Reading Material

Links to the units mentioned in the slides, and additional materials.

5) Questions and Answers

FAQs that students have following the lecture or instructions.
15.1. *Dramatis personae*
These are the TAs.

At ETHZ:
- Shiying Li (shili@student.ethz.ch)
- Ercan Selçuk (ercans@student.ethz.ch)
- Miguel de la Iglesia Valls (dmiguel@student.ethz.ch)
- Khurana Harshit (hkhurana@student.ethz.ch)
- Lapandic Dzenan (ldzenan@student.ethz.ch)
- Marco Erni (merni@ethz.ch)

At TTIC:
- Andrea F. Daniele (afdaniele@ttic.edu)
- Falcon Dai (dai@ttic.edu)
- Jon Michaux (jmichaux@ttic.edu)

At Montreal:
- Florian Golemo (fgolemo@gmail.com)

15.2. *First steps*
Here are the first steps for the TAs.
Note that many of these are not sequential and can be done in parallel.

1) **Learn about Duckietown**

Read about Duckietown’s history; watch the Duckumentary.

   → *Unit A-1 - The Duckietown project*

2) **Online accounts**

You have to set up:
- A personal Github account
- A Twist account
- A Slack account
- A Google Docs account (Gmail address)

Send an email to Kirsten Bowser (akbowser@gmail.com), with your GMail address and your Github account. She will give you further instructions.

**Point of contact:** Kirsten Bowser

3) **Install Ubuntu**
Install Ubuntu 16.04 on your laptop, and then install ROS, Atom, Lclipse, etc.

4) Duckuments

Install the Duckuments system, so you can edit these instructions.

- Duckumentation documentation.

Point of contact: Andrea

5) Learn about Git and Github

Start learning about Git and Github. You don’t have to read the entirety of the following references now, but keep them “on your desk” for later reference.

- Good book
- Git Flow

Point of contact: Liam?

6) Continuous integration

Understand the continuous integration system.

- Documentation on continuous integration.

Point of contact: Andrea

7) Duckiebot building

Build your Duckiebot according to the instructions.

- Part B - Duckiebot assembly and setup

Point of contact: Shiyeng (ETH)

<table>
<thead>
<tr>
<th>Point of contact: ??? (UdeM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Point of contact: ??? (TTIC)</td>
</tr>
</tbody>
</table>

As you read the instructions, keep open the Duckuments source, and note any discrepancies. You must note any unexpected thing that is not predicted from the instruction. If you don’t understand anything, please note it.
The idea is that dozens of other people will have to do the same after you, so improving the documentation is the best use of your time, and it is much more efficient than answering the same question dozens of times.

8) Other documentation outside of the Duckuments

We have the following four documents outside of the duckuments:

1. **Organization chart**: This is where we assign areas of responsibility.
2. **Lecture schedule**
3. **Checkoff spreadsheet**
4. **The big TODO list**: Where we keep track of things to do.
UNIT A-16

Checkoff: Assembly and Configuration

The first job is to get your Duckiebot put together and up and running.

16.1. Pick up your Duckiebox

Slack channel: #help-parts

There is a checklist inside. You should go through the box and ensure that the parts that are supposed to be in it actually are inside.

If you are missing something, contact your local responsible and ask for help on Slack in the appropriate channel.

These sections describe the parts that are in your box.

16.2. Soldering your boards

Depending on how kind your instructors/TAs are, you may have to solder your boards.

You don’t need to buy anything - you have all the parts that you need.
16.3. Assemble your Robot

You don’t need to solder anything.

Slack channel: #help-assembly

You are ready to put things together now.
16.4. Optional: Reproduce the SD Card Image
If you are very inexperienced with Linux/Unix/networking etc, then you may find it a valuable experience to reproduce the SD card image to “see how the sausage is made”.

You probably don’t want to see how the sausage is made.

16.5. Setup your laptop
Slack channel: #help-laptops
The only officially supported OS is Ubuntu 16.04. If you are not running this OS it is recommended that you make a small partition on your hard drive and install the OS.
Related parts of the book are:
16.6. Make your robot move
Slack channel: #help-robot-setup
Now you need to clone the software repo and run things to make your robot move. First initialize the robot:

- Unit B-5 - Duckiebot Initialization

Then get it to move!

- Unit B-8 - Making your Duckiebot move)

16.7. Upload your video
You should record a video demonstrating that your Duckiebot is up and running. Brownie points for creative videos. Please upload your videos via the following URL:

?? | Chicago: upload your video
?? | Zurich: upload your video
UNIT A-17

Checkoff: Take a Log

KNOWLEDGE AND ACTIVITY GRAPH

| Requires: Unit A-16 - Checkoff: Assembly and Configuration |
| Results: A verified log in rosbag format uploaded to Dropbox. |

Slack channel: #help-logging

?? | Montreal deadline: Oct 4, 11pm
?? | Zurich deadline: Oct 20, 17:00

17.1. Mount your USB drive
We will log to the USB drive that you were given.

- Unit A-16 - Mounting USB drives

17.2. Take a Log
Take a 5 min log as you drive in Duckietown.

?? | For Montreal this is rm. 2333.
?? | For Zurich this is ML J44. Ask the TA when it is available.
?? | For Chicago, we are still building the town, so feel free to do this at home or in the lab.

- Unit B-12 - Taking and verifying a log for detailed instructions.

17.3. Verify your log

- Section 12.5 - Verify a log for detailed instructions.

17.4. Upload the log

?? | Upload the log here
?? | Upload the log here
?? | Upload the log here
UNIT A-18

Checkoff: Robot Calibration

KNOWLEDGE AND ACTIVITY GRAPH

| Requires: | Unit A-16 - Checkoff: Assembly and Configuration |
| Requires: | That you have correctly cloned and followed the git procedure outline in Unit A-7 - Git usage guide for Fall 2017. |
| Requires: | That you have correctly setup your environment variables according to (unknown ref software_devel/env-variables) |

Results: You robot calibrations (wheels and camera (x2)) are merged to git through a PR.

Slack channels: #help-wheel-calib, #help-camera-calib

18.1. Pull and rebuild your Software repo on robot and laptop

Some of the services have changed and this requires a rebuild.

On both laptop and robot do:

```bash
$ cd Duckietown root
$ source environment.sh
$ make build-catkin-clean
$ make build-catkin-parallel
```

18.2. Make a branch in the duckiefleet repo

Remember that the git policy has changed a bit. You are probably best to re-clone the duckiefleet repo. For details see Unit A-7 - Git usage guide for Fall 2017 and particularly the section For U de M students who have already submitted homework to the previus duckiefleet-2017 repo

Don't forget that master is now protected in duckiefleet. So make a new branch right away and call it `GIT_USERNAME -devel`
18.3. Kinematic calibration
Follow the procedure in Unit B-11 - Wheel calibration. Once you have successfully passed the automated test, take a screen shot and post it to the slack channel #checkoffs and we will all congratulate you.

18.4. Camera calibration
Follow the procedure in Unit B-10 - Camera calibration and validation to do you intrinsic and extrinsic calibrations.

18.5. Visually verify the calibration is good in Duckietown
Take your robot to Duckietown. Put it in a lane.

On your robot execute

```
$ make demo-lane-following
```

On your laptop do (after setting ros master to your robot):

```
$ rqt_image_view
```

on your joystick you need to hit the top-right button (TODO: add picture). On the command line you should see the output `state_verbose = True`
in the drop down menu select `robot name /line_detector_node/image_with_lines`
on the display you should see all the color-coded line detections

now open the Rviz visualizer on your laptop (after setting ros master to your robot):

```
$ rviz
```

• click the Add button in the bottom left.
• then click the By Topic tab
• then click the triangle next to `/segment_list_markers` underneath `/duckiebot_visualizer`
  • then double click on MarkerArray

On the display you should see the ground projected lines. Do they make sense? If not your calibration is wrong.

18.6. Submit a PR
Don’t forget at the end to submit a PR back to duckiefleet repo
### Unit A-19

**Homework: Data Processing (UdeM)**

**Knowledge and Activity Graph**

| Requires: Unit A-17 - Checkoff: Take a Log |
| Requires: Unit E-1 - ROS installation and reference |
| Results: Ability to perform basic operations on images |
| Results: Build your first ROS package and node |
| Results: Ability to process imagery live |

**Slack channel:** #ex-data_processing

**Montreal deadline:** Oct 4, 11:00pm

19.1. **Follow the git policy for homeworks**
   - Section 7.3 - Git policy for homeworks (TTIC/UDEM)

   for instructions on how the homework should be submitted.

19.2. **Exercise: Basic image operations**

   Complete Unit A-2 - Exercise: Basic image operations, adult version

19.3. **Exercise: Log decimation**

   Complete Unit A-4 - Exercise: Bag in, bag out

19.4. **Exercise: Instagram filters**

   Complete Unit A-6 - Exercise: Instagram filters

19.5. **Exercise: Live Instagram**

   Complete Unit A-8 - Exercise: Live Instagram

   Call your package `dt-instagram-live_ robot name` and call your node `dt-instagram-live_ robot name`

   When you are done, take a 5min log (See Unit B-12 - Taking and verifying a log) in Duckietown (2333 in Montreal) and upload [here](#)
UNIT A-20

Homework: Data Processing (TTIC)

KNOWLEDGE AND ACTIVITY GRAPH

| Requires: Unit E-1 - ROS installation and reference |
| Results: Ability to perform basic operations on images |
| Results: Build your first ROS package and node |
| Results: Ability to process imagery live |

Slack channel: #help-data-processing

TTIC deadline: Friday, October 13 11:59pm CT

20.1. Follow the git policy for homeworks

   - Section 7.3 - Git policy for homeworks (TTIC/UDEM)

for instructions on how the homework should be submitted.

20.2. Exercise: Basic image operations

Complete Unit A-2 - Exercise: Basic image operations, adult version

20.3. Exercise: Log analysis

Complete Unit A-3 - Exercise: Simple data analysis from a bag

20.4. Exercise: Log decimation

Complete Unit A-4 - Exercise: Bag in, bag out

20.5. Exercise: Video thumbnails

Complete Unit A-5 - Exercise: Bag thumbnails

20.6. Exercise: Instagram filters

Complete Unit A-6 - Exercise: Instagram filters

20.7. Exercise: Log Instagram

Complete Unit A-7 - Exercise: Bag instagram

20.8. Exercise: Live Instagram
Complete Unit A-8 - Exercise: Live Instagram
Call your package `dt-instagram-live_robot name` and call your node `dt-instagram-live_robot name`.

20.9. Exercise: Feedback
Complete the exercise feedback form. You will receive points towards this exercise if you complete the form.
UNIT A-21

Exercises: Data Processing (Zurich)

**KNOWLEDGE AND ACTIVITY GRAPH**

- **Requires:** Unit E-1 - ROS installation and reference
- **Results:** Ability to perform basic operations on images
- **Results:** Build your first ROS package and node
- **Results:** Ability to process imagery live

Slack channel to get help: #ex-data-processing

21.1. Git setup

First, make sure you are in the Github Zurich team.
If your name is not here, contact Kirsten, and stop. You will not be able to do the next step.
Please clone this repository:

```bash
$ git clone git@github.com:AndreaCensi/exercises-fall2017.git
```

This repository is writable by all Zurich people, but not readable by Chicago and Montreal. Because they grade the homework, we need to keep it secret.
We invite everybody to just push their exercises to this repository. (This is also compulsory to get help from TAs, so that the TAs can give comments that are useful for everybody.)

21.2. The exercises

We have created a series of exercises that are supposed to help somebody who doesn’t know how to program in Python/Linux to get to a decent level.
We suggest the following:
1. First, read through the exercises and note the skills that are learned for each one.
2. Look at the last two: “Log Instagram” and “Live Instagram”. Do you think you can do them? If so, just do those two and feel free to skip the rest.
3. Otherwise, you have a lot to catch up. No problem. Take your time. Start with the basic exercise. TAs are here to help.

21.3. Exercise: Basic image operations
1) Exercise: Log analysis
- Unit A-2 - Exercise: Basic image operations, adult version

2) Exercise: Log decimation
- Unit A-3 - Exercise: Simple data analysis from a bag

3) Exercise: Video thumbnails
- Unit A-4 - Exercise: Bag in, bag out

4) Exercise: Instagram filters
- Unit A-5 - Exercise: Bag thumbnails

5) Exercise: Log Instagram [recommended for everybody]
- Unit A-6 - Exercise: Instagram filters

6) Exercise: Live Instagram [recommended for everybody]
- Unit A-7 - Exercise: Instagram [recommended for everybody]

Call your package `dt-instagram-live_robot name` and call your node `dt-instagram-live_robot name`.

7) Exercise: Feedback form
UNIT A-22

Homework: Augmented Reality

KNOWLEDGE AND ACTIVITY GRAPH

| Requires: Unit E-1 - ROS installation and reference |
| Requires: Unit B-10 - Camera calibration and validation |
| Results: Ability to project fake things from an image back into the world |

Slack channel: #ex-augmented_reality

- Montreal deadline: Oct 27, 11:00pm
- Chicago deadline: Oct 27, 11:59pm
- Zurich deadline: Oct ???, 11:59pm

22.1. Follow the git policy for homeworks

Please follow the instructions on how the homework should be submitted.

- Section 7.3 - Git policy for homeworks (TTIC/UDEM)

22.2. Exercise: Augmented Reality

Complete Unit A-9 - Exercise: Augmented Reality.

- Please hold on for Github instructions.

Note that you should do a pull in Software to get all the goodies and utils described in the exercise, and exercises-fall2017 repos (in exercises-fall2017 repo this means pulling from the duckietown remote):

```bash
$ git pull upstream master
```

if you have followed the instructions properly.

In the exercises-fall2017 repository, you will find a template that you can use to make your own package. Basically everywhere you see littleredcorvette you would need to replace it with your robot_name.

22.3. Submission

- Please upload the images requested in the homework to your git repository. When complete, please tag a release from your repo.

- Please upload the images requested in the homework to your git repository. When complete, please tag a release from your repo.
22.4. **Bonus: Defining intersection_4way.yaml**

The first student to do it (from any institution) gets notoriety and a bonus.
UNIT A-23
Checkoff: Navigation

**Knowledge and Activity Graph**

| Requires: Unit A-18 - Checkoff: Robot Calibration |
| Requires: Unit A-17 - Checkoff: Take a Log |
| Results: 2 logs of your robot autonomously navigating Duckietown |

?? | Montreal Deadline: Nov 15, 11pm

?? | Chicago Deadline: Nov 15, 11pm

Slack channel: [#help-navigation](https://slack.com)

---

23.1. Pull from master

As always - it’s a good idea to pull from `master` often.

23.2. Lane Following

Place your robot on the Duckietown map somewhere on the “outer loop” (right hand lane so that it will follow the exterior of the map).

Launch the robot with the command from `DUCKIETOWN_ROOT`:

```
$ make demo-lane-following
```

Open a terminal on your laptop and set the ros master to your robot.

Toggle the `VERBOSE` flag by writing:

```
$ rospam set /robot_name/line_detector_node/verbose true
```

Then open `rqt_image_view`. Look at the `../image_with_lines` image output. Apply the anti-instagram calibration by pushing the `Y` button on the joystick (TODO: is it the same for the new joysticks?). You should see your image get corrected and the line detections become more correct. If nothing happens and your robot output complains of bad health, move the robot a little bit and try again.

You may also be interested to look at the `../belief_img` to see the output of the histogram filter. It should be quite stable if your robot is not moving. You can move the robot around to see how the posterior is updating.

If everything is looking good then push the `START` button on the joystick and your robot should start to drive.

The robot operation should look like [this](#).

Follow the instructions [here](#) to take a **minimal** log of at least 5 mins of uninterrupted
robot autonomous function.

1) Bonus

The student who uploads the longest log of uninterrupted robot autonomous lane following from any institution will get a great bonus.

23.3. Indefinite Navigation

Follow the exact same procedure above but instead of running the lane following demo run the “indefinite navigation” demo:

$ make indefinite-navigation

Your robot will now stop at the stop lines and then make a random turn through the intersection. If it is crashing a lot you may need to turn the trajectories it takes through the intersection. To do so you may need to edit the file here:

```text
turn_left: #time, velocity, angular vel
  - [0.8, 0.43, 0]
  - [1.8, 0.43, 2.896]
  - [0.8, 0.43, 0.0]

turn_right:
  - [0.6, 0.43, 0]
  - [1.2, 0.3, -4.506]
  - [1.0, 0.43, 0.0]

turn_forward:
  - [0.8, 0.43, 0.4]
  - [1.0, 0.43, 0.0]
  - [1.0, 0.43, 0.0]
```

to make it more reliably traverse the intersections.

Follow the instructions here to take a minimal. You may use the BACK button to stop it from crashing and then return it to autonomous mode with the START button.

1) Bonus

The student who uploads the longest log of uninterrupted robot autonomous indefinite navigation from any institution will get a great bonus.
UNIT A-24
Homework: Lane Filtering

Montreal deadline: Nov 17, 11:00pm
Chicago deadline: Nov 17, 11:59pm

Slack channel: #ex-filtering

24.1. Follow the git policy for homeworks
Please follow the instructions on how the homework should be submitted.
- Section 7.3 - Git policy for homeworks (TTIC/UDEM)

24.2. Pick your Poison
This homework is about filtering. Either replace the existing histogram lane filter with either an Extended Kalman Filter or a Particle Filter. If you do both you will get a bonus.

24.3. Setup instructions
Pull from master in the Software repo
Pull from the Duckietown (upstream) remote in your exercises-fall2017 repo.
We are providing a script to change all the instances of the default robot (in this case shamrock) with YOUR_ROBOT_NAME to save you time. To run navigate to the homeworks/03_filtering directory and run:

```
$ ./change_robot_name_everywhere.sh YOUR_ROBOT_NAME
```

(you're welcome...)

In the

```
`homeworks/03_filtering/ YOUR_ROBOT_NAME /dt_filtering_ YOUR_RO-
BOT_NAME`
```

folder, the files you need to worry about are the following:
1) default.yaml: this contains the parameters that will be loaded. Here’s what it currently looks like:
# default parameters for lane_filter/lane_filter_node
# change to your robot name below
filter:
 - dt_filtering_shamrock.LaneFilterPF
   - configuration:
     example1: 0.2
     # fill in params here

# uncomment below and comment above if you are doing EKF
# - dt_filtering_shamrock.LaneFilterEKF
# - configuration:
#   example2: 0.3
# #fill in other params here

This parameter file tells your node to automatically load the right filter. If you are working on particle filter you can leave it the way it is and just add your parameters that you need under configuration. If you are working on EKF, comment or delete the lines for the PF and uncomment the lines for the EKF and then add your params as needed.

The other file you need to concern yourself with is in

```
include/dt_filtering_ YOUR_ROBOT_NAME
```

You will need to fill in the functions that are setup for you.

### 24.4. Submission

As normal, tag the TAs and instructors in a release from your repo when you are ready for your work to be evaluated.
UNIT A-25
Guide for mentors

Assigned to: Liam
Unit A-26
Project proposals
27.1. Preliminaries

**Name of Project:** System Architecture

**Slack channel:** #devel-heroes

**Software development branch:** devele-sonja (Software repo) or sonja-branch (Duckuments repo)

1) Missions

The system architect project can be split into two missions:

1. Ensure that the development of the system goes smoothly (wooden spoon)
2. Develop a framework/tool to formally describe (and later optimize) the system (brass, silver and gold)

27.2. Mission 1

Ensure that the development and integration of the projects into the system goes smoothly and that the resulting system makes sense, and is useful for future duck-iterations (duckie + generations).

1) Problem Statement

Ensure that all teams know what their goal is and how it fits into the bigger picture

2) Relevant Resources

- The functional diagram of the system
- Duckuments
- Other teams’ preliminary project reports

3) Deliverables (Goals)

The deliverables for Mission 1 will include the following:

- Functional diagram of the system
- Documentation of system architecture

Mission 1 is the “wooden spoon” level of the project.

4) Proposed Approach

- Become one with the goals of Duckietown In order to make Duckietown a better place, one has to keep in mind what “better” means in Duckie terms.
- Be familiar with the current system architecture and track changes This can include having to update the functional diagram, for instance.
- Keep in close contact with teams This will be done by attending the meetings
of some of the other teams (especially early meetings). Some teams’ meetings have been prioritized since many parts of the system are dependant on their work, namely:
  - Anti-instagram
  - Controllers
  - Navigators

  - Explicit coordination All teams will designate a contact person who can contact me whenever they change their project boundaries or have doubts/ need advice on their project’s boundaries/negotiating with other

- Offer nudges in a different direction if needed
- Acting as middleman/helper to facilitate negotiation of contracts between groups
- Monitor status of projects to find possible problems

5) Logging and Testing Procedure

... 

6) Current status

Familiarisation with the current system status is under way.
Functional diagram has been updated to include multi-robot SLAM as alternative to single-robot SLAM to creating map.

7) Tasks

- Familiarisation with existing system architecture
- Going to group meetings
- Identifying potential problems

8) Timeline

... 

9) Meetings notes

...

27.3. Mission 2

Where there is a system, there is a want (nay, need) for optimisation. Describing a system’s performance and resource requirements in a quantifiable way is a critical part of being able to benchmark modules and optimise the system.

Mission 2 is to formalise the description of the system characteristics, so that eventually the system performance can be optimised for some given resources.

1) Problem Statement

Find a way to describe all the module requirements, specifications, etc in a formal, quantifiable language.
Find a way to calculate the requirements and specifications of a whole system or subsystem, based on the requirements and specifications of the individual modules of the system.

Find a way to calculate the optimal system configuration, based on the desired requirements and specifications of the system.

2) Relevant Resources

- How the current system’s characteristics are defined
- Which values/parameters are needed
- Possibly research on system description?
- Possibly graph theory?

3) Deliverables (Goals)

The different levels of Mission 2 are defined as follows:

- Bronze standard:
  - Formal, qualitative language to describe constraints/requirements between modules
- Silver standard:
  - Each module has a table of performance. Qualitative. Can compare and give yes/no queries. With given configuration x, is the cost/requirements possible with available resources? f(x) smaller equal to Rmax?
- Gold standard:
  - Optimization is possible to find the best implementation, given available resources. Given f(x) and Rmax, find optimal configuration x

The deliverables will then include:

- Documentation on the result of the project
- A description of the current system’s characteristics (bronze)
- A program/tool that can give a qualitative answer (yes/no) to the question: Are these resources sufficient for this system configuration? (silver)
- A program/tool that will give an optimised system configuration, based on the given available resources (gold)

4) Proposed Approach

- Research on the topic of formal description of a system
- Find/develop a suitable language to describe module characteristics
- Require groups to compile a description of their respective modules’ characteristics
- Find/develop functions to do mathematics on the language description of modules

5) Logging and Testing Procedure

...
6) Current status
Research is being done to identify some research areas that may be relevant and tools that may be helpful, in order to decide on an approach.

7) Tasks
- Research into existing methods of system description
- Graph based databases?
- Perhaps graph theory can be useful later if the (suspiciously graph-looking) system can be described suitably.

8) Timeline
...

9) Meetings notes
...

UNIT A-28

Template of a project

Make a copy of this document before editing.

28.1. Preliminaries
Name of Project:
Team:
• Person 1 (UdeM) (Mentor)
• Person 2 (TTIC) (TA)
• ...

| Slack channel: #XXX |
| Software development branch: XXX-devel |

28.2. Problem Statement
Summarize the mission for the team - What is the need that is being addressed? Do not focus on technical specifics yet.

28.3. Relevant Resources
List papers, open source code, pages in the Duckiebook, lecture slides, etc, that could be relevant in your quest.

28.4. Deliverables (Goals)
Anything that is going to be an output. These should be quantified in terms of functionalities and performance metrics where appropriate.
Example 1: A Duckiebot detection system (functionality) with minimum precision of 0.8, a minimum recall of 0.5, a maximum latency of 50ms with maximum CPU consumption of 80% of one core (performance).
Example 2: At least 20 hours of logs Duckiebots annotated
• Bronze standard:
• Functionality:
• Performance:
• Silver standard:
• Functionality:
• Performance:
• Gold standard:
• Functionality:
• Performance:
Part of the deliverables should be:
1. A new or improved functionality (demonstrated live and with a video) with well-
documented code,
2. A approx 15-20min presentation about the functionality (with a slide deck + 1
slide overview poster to be shown at public demo),
3. A technical description of the underlying method in the form of a page in the
Duckiebook,
4. Instructions for reproducing the functionality in the form of a page in the duck-
iebook.

28.5. Proposed Approach
After analysis of the resources and precise understanding of the problem you trying
to solve, make a plan for how you will solve the problem. It is possible that at the
start you could explore several seemingly promising avenues. However, you should
converge on Bronze standard before moving to Silver standard etc.
• Bronze standard:
• Silver standard:
• Gold standard:

28.6. Logging and Testing Procedure
A detailed description of the logs and procedure you will use to verify that the system
is working the way you say it is working. In most cases this should include a regression
test so that when someone changes something else, we can make sure that your
thing still works as well as it used to.

28.7. Current status
Write here the current status. What works now, as opposed to what the goal is. The
difference between these two is the work to be done.

Note: it is better to have something that does not work, and a good description
of what should work and why it doesn't work, than to have something that kinda
works, but nobody knows what the thing is supposed to do.

1) Functionality
Nothing implemented.

2) Performance
Infinitely slow.

28.8. Tasks
So and so should do such and such
28.9. Timeline
- This should be done by Nov 15
- That should be done by Nov 16

28.10. Meetings notes
- Link 1
- Link 2
UNIT A-29
The Map Description

The map to be used in the Fall 2017 class is shown in Figure 29.2.

Figure 29.2. The map to be used in the Fall 2017 class
The editable keynote file is in this directory of the duckuments repo. The ids on the signs correspond to the Apriltag IDs. For more details see (unknown ref opmanual_duckietown/signage)

I will ignore this because it is an external link.
> I do not know what is indicated by the link '#opmanual_duckietown/signage'.