



The Duckietown Book



The last version of this book and other documents are available at the URL
<http://book.duckietown.org/>

TABLE OF CONTENTS

Part 1 - The Duckietown project..... 4

Chapter 1 - What is Duckietown?..... 5

 Section 1.1 - Goals and objectives..... 5

 Section 1.2 - Results obtained so far 5

 Section 1.3 - Why the duckies? 5

 Section 1.4 - Learn about the platform..... 5

 Section 1.5 - Learn about the educational experience 6

 Section 1.6 - Learn about the platform..... 6

Chapter 2 - Duckietown history and future..... 7

 Section 2.1 - The beginnings of Duckietown 7

 Section 2.2 - Duckietown around the world..... 7

 Section 2.3 - Coming up..... 7

Chapter 3 - First steps 8

 Section 3.1 - How to get started 8

 Section 3.2 - How to keep in touch 8

 Section 3.3 - How to contribute 8

Chapter 4 - Duckietown for instructors 9

Chapter 5 - Duckietown for self-guided learners..... 10

Chapter 6 - Introduction for companies 11

Chapter 7 - Frequently Asked Questions..... 12

 Section 7.1 - General questions..... 12

 Section 7.2 - FAQ by students / independent learners 12

 Section 7.3 - FAQ by instructors 12

 Section 7.4 - FAQ by researchers 12

Part 2 - Setup instructions 13

Chapter 8 - Acquiring the parts for the Duckiebot 14

 Section 8.1 - Bill of materials..... 14

 Section 8.2 - Chassis..... 14

 Section 8.3 - Raspberry PI 3..... 14

 Section 8.4 - Camera 15

Chapter 9 - Soldering boards..... 16

Chapter 10 - Assembling the Duckiebot..... 17

Chapter 11 - Reproducing the Ubuntu image for the Duckiebot 18

 Section 11.1 - Step 1: From blank to minimal setup 18

Chapter 12 - Reproducing the Ubuntu image for laptops..... 20

 Section 12.1 - Useful software..... 20

 Section 12.2 - Installation of the documentation system 21

Chapter 13 - Burn the image..... 22

Chapter 14 - Setup Buffalo access point..... 23

Chapter 15 - Setup the laptops..... 24

Chapter 16 - Github basics 25

 Section 16.1 - Get an account 25

 Section 16.2 - Create an SSH Key on your machine 26

Chapter 17 - SD Image to remote control 28

Chapter 18 - RC remote control..... 29

Chapter 19 - Joystick plus camera output..... 30

Chapter 20 - Wheel calibration 31

Chapter 21 - Camera calibration..... 32

Chapter 22 - Taking a log..... 33

Chapter 23 - Create a ROS package..... 34

Chapter 24 - Reference for useful software commands..... 35

 Section 24.1 - SSH..... 35

Section 24.2 - Byobu	35
Section 24.3 - VIM	35
Section 24.4 - (http)	35
Section 24.5 - Git and related commands	35
Chapter 25 - Integrate package in the architecture.....	36
Chapter 26 - LED Setup.....	37
Chapter 27 - Acquiring the parts for Duckietown (2017)	38
Chapter 28 - Traffic Light Assembly.....	39
Part 3 - Modeling.....	40
Chapter 29 - Kinematics of Duckiebot.....	41
Part 4 - How to contribute	42
Chapter 30 - Accounts	43
Section 30.1 - Complete list of accounts.....	43
Section 30.2 - For Fall 2017	43
Section 30.3 - For other contributors.....	43
Chapter 31 - Contributing to the documentation	44
Section 31.1 - Where the documentation is	44
Section 31.2 - Editing links.....	44
Section 31.3 - Comments	44
Section 31.4 - Installing dependencies for compiling the documentation	44
Section 31.5 - Troubleshooting installation problems	45
Section 31.6 - Compiling the documentation	45
Section 31.7 - Deploying the documentation	46
Chapter 32 - Features of the documentation writing system.....	47
Section 32.1 - Embedded LaTeX	47
Section 32.2 - Other interesting features.....	47
Section 32.3 - Limitations.....	47
Chapter 33 - Bibliography.....	48

PART 1

The Duckietown project

CHAPTER 1

What is Duckietown?

1.1. Goals and objectives

Duckietown is a robotics education and outreach effort.

The most tangible goal of the project is to provide a low-cost educational platform for learning autonomy, consisting of the Duckiebots, an autonomous robot, and the Duckietowns, the infrastructure in which the Duckiebots navigate.

However, we focus on the *learning experience* as a whole, by providing a set of modules teaching plans and other guides, as well as a curated role-play experience.

We have two targets:

1.

For **instructors**, we want to create a “class-in-a-box” that allows to offer a modern and engaging learning experience. Currently, this is feasible at the advanced undergraduate and graduate level, though in the future we would like to present the platform as multi-grade experiences.

2.

For **self-guided learners**, we want to create a “self-learning experience”, that allows to go from zero knowledge of robotics to graduate-level understanding.

In addition, the Duckietown platform has been used as a research platform.

TODO: add references to papers submitted/published with it.

1.2. Results obtained so far

While we are at the early phases of the project, many people have been using the materials in the past year.

1.3. Why the duckies?

Why the duckies?

Compared to other educational robotics projects, the presence of the duckies is what makes this project stand out. Why the duckies?

We want to present robotics in an accessible and friendly way.

TODO: copy usual discussion from somewhere else.

TODO: add picture of kids with Duckiebots.

1.4. Learn about the platform

The best way to get a sense of how the platform looks is to watch these videos. They show off the capabilities of the platform.

TODO: add 4 “cool” videos currently in the home page

This video is part of the Red Hat documentary:

TODO: add Red Hat video

1.5. Learn about the educational experience

These papers present a more formal description of the technical side of the project as well as the educational side.

This paper [\[1\]](#) describes the course design for Duckietown: learning objectives, teaching methods, etc.

This video is a Duckumentary about the first version of the class, during Spring 2016. The Duckumentary was shot by Chris Welch.

TODO: Duckumentary about the release.

1.6. Learn about the platform

The paper [\[2\]](#) describes the Duckiebot and its software. With 29 authors, we made the record for a robotics conference.

CHAPTER 2

Duckietown history and future

2.1. The beginnings of Duckietown

Duckietown started as an MIT class during Spring 2016.

2.2. Duckietown around the world

1) Duckietown High School

2.3. Coming up

In 2017, the class will be offered contemporaneously at:

- ETH Zurich
- University of Montreal
- University of Chicago

as well as:

CHAPTER 3

First steps

3.1. How to get started

If you are an instructor, please jump to [Chapter 4](#).

If you are a self-guided learner, please jump to [Chapter 5](#).

If you are a company, and interested in working with Duckietown, please jump to [Chapter 6](#).

3.2. How to keep in touch

TODO: add link to Facebook

TODO: add link to Mailing list

TODO: add link to Slack?

3.3. How to contribute

TODO: If you want to contribute to the software...

TODO: If you want to contribute to the hardware...

TODO: If you want to contribute to the documentation...

TODO: If you want to contribute to the dissemination...

CHAPTER 4

Duckietown for instructors

CHAPTER 5

Duckietown for self-guided learners

TODO: to write

CHAPTER 6

Introduction for companies

TODO: to write

CHAPTER 7

Frequently Asked Questions

7.1. General questions

What is Duckietown?

Duckietown is a low-cost educational and research platform.

Is Duckietown free to use?

Yes. All materials are released according to an open source license.

Is everything ready?

Not quite! Please [sign up to our mailing list](#) to get notified when things are a bit more ready.

How can I start?

See the next section, Getting started.

How can I help?

If you would like to help actively, please email duckietown@mit.edu.

7.2. FAQ by students / independent learners

I want to build my own Duckiebot. How do I get started?

TODO: to write

7.3. FAQ by instructors

How large a class can it be? I teach large classes.

TODO: to write

What is the budget for the robot?

TODO: to write

I want to teach a Duckietown class. How do I get started?

Please get in touch with us at duckietown@mit.edu. We will be happy to get you started and sign you up to the Duckietown instructors mailing list.

7.4. FAQ by researchers

TODO: to write

PART 2

Setup instructions

CHAPTER 8

Acquiring the parts for the Duckiebot

The trip begins with acquiring the parts. Here, we provide a link to all bits and pieces that are needed to build a Duckiebot and the Duckietowns.

In general, keep in mind that:

- The links might expire, or the prices might vary.
- In general, substitutions are OK for the mechanical components, and not OK for all the rest.

8.1. Bill of materials

Chassis	USD xxx
Camera	USD xxx
Raspberry PI 3	USD 35
Total for minimum configuration	USD ??
Total for fancy configuration	USD ??

8.2. Chassis

We selected the Magician Chassis as the basic chassis for the robot (Figure 1). We chose it because it has a double-decker configuration, and so we can put the battery in the lower part. The price for this in the US is about USD 15-30.



Figure 1. The Magician Chassis

8.3. Raspberry PI 3

...

8.4. Camera



...

CHAPTER 9

Soldering boards

CHAPTER 10

Assembling the Duckiebot



CHAPTER 11

Reproducing the Ubuntu image for the Duckiebot

These are the instructions to reproduce the Ubuntu image that we use.

We organize this in three steps:

- Step 1: From downloaded image, to updated setup.
- Step 2: Installation of ros and other dependencies.

11.1. Step 1: From blank to minimal setup

Resources necessities:

- Internet connection to download the packages.
- A PC running any Linux with an SD card reader.
- Time: about 20 minutes.

Results:

- A baseline Ubuntu Mate 16.04.2 image with updated software.

1) Download and uncompress the Ubuntu Mate image

Download the image from the page <https://ubuntu-mate.org/download/>.

The file we are looking for is:

```
filename: ubuntu-mate-16.04.2-desktop-armhf-raspberry-pi.img.xz
size: 1.2 GB
SHA256: dc3afcad68a5de3ba683dc30d2093a3b5b3cd6b2c16c0b5de8d50fede78f75c2
```

Run the command `sha256` to make sure you have the right version:

```
laptop $ sha256 ubuntu-mate-16.04.2-desktop-armhf-raspberry-pi.img.xz
dc3afcad68a5de3ba683dc30d2093a3b5b3cd6b2c16c0b5de8d50fede78f75c2
```

If the string does not correspond exactly, your download was corrupted. Delete the file and try again.

Then decompress using the command `xz`:

```
laptop $ xz -d ubuntu-mate-16.04.2-desktop-armhf-raspberry-pi.img.xz
```

2) Finding your device name for the SD card

TODO: How to find the device name?

3) Burn the image to an SD card

Then burn to disc using the command `dd`:

```
laptop $ sudo dd of=DEVICE if=IMG status=progress bs=4M
```

where `IMG` is the `.img` file you unzipped, and `DEVICE` is the device that represents your SD card reader.

4) Verify that the SD card was created correctly

Remove the SD card and plug it in again in the laptop.

Ubuntu will mount two partitions, by the name of `PI_ROOT` and `PI_BOOT`.

5) Installation

Boot the disk in the Raspberry PI.

I chose the following options:

```
language: English
username: ubuntu
password: ubuntu
hostname: duckiebot
```

Then I rebooted.

6) Update installed software

The WiFi was connected to airport network `duckietown` with password `quackquack`.

Afterwards I upgraded all the software preinstalled with these commands:

```
duckiebot $ sudo apt-get update
duckiebot $ sudo apt-get dist-upgrade
```

CHAPTER 12

Reproducing the Ubuntu image for laptops

These are Andrea's notes for the laptops (Duckietops).

The image was Ubuntu Mate 16.04.2.

I chose the following options:

```
language: English
username: ubuntu
password: ubuntu
hostname: duckietop
```

If you choose a different username, you will need to change all the commands later.

12.1. Useful software

Use `etckeeper` to keep track of the configuration in `/etc`:

```
laptop $ sudo apt install etckeeper
```

Install `ssh` to login remotely:

```
laptop $ sudo apt install ssh
```

Use `byobu`:

```
laptop $ sudo apt install byobu
```

Use `vim`:

```
laptop $ sudo apt install vim
```

Use `htop` to monitor CPU usage:

```
laptop $ sudo apt install htop
```

Additional utilities for `git`:

```
laptop $ sudo apt install git-extras
```

Other utilities:

```
laptop $ sudo apt install avahi-utils ecryptfs-utils
```

1) Redshift

This is Flux for Linux. It is an accessibility/lab safety issue: bright screens damage eyes and perturb sleep [3].

Install `redshift` and run it.

```
laptop $ sudo apt install redshift-gtk
```

Set to “autostart” from the icon.

12.2. Installation of the documentation system



Next, the docs system was installed using the documentation in [Section 31.4](#).

CHAPTER 13

Burn the image

CHAPTER 14

Setup Buffalo access point

CHAPTER 15

Setup the laptops

CHAPTER 16

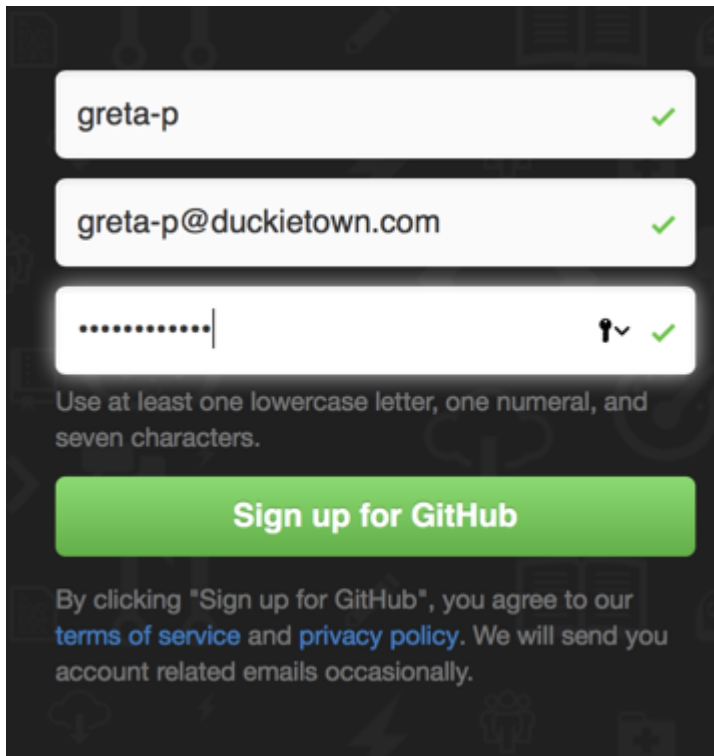
Github basics

16.1. Get an account

Our example account is the following:

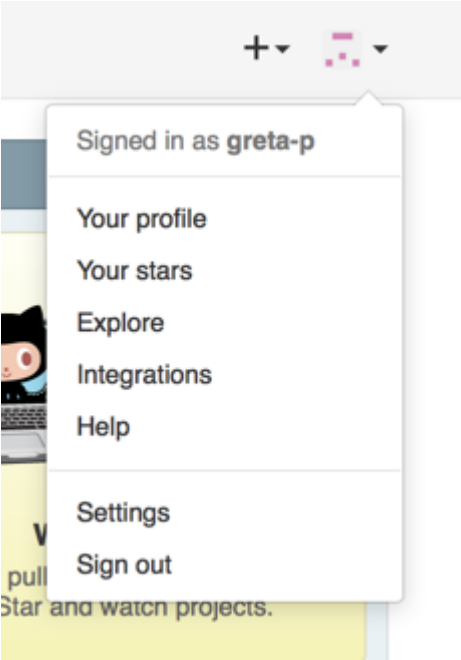
Github name: greta-p
E-mail: greta-p@duckietown.com

Create a Github account.

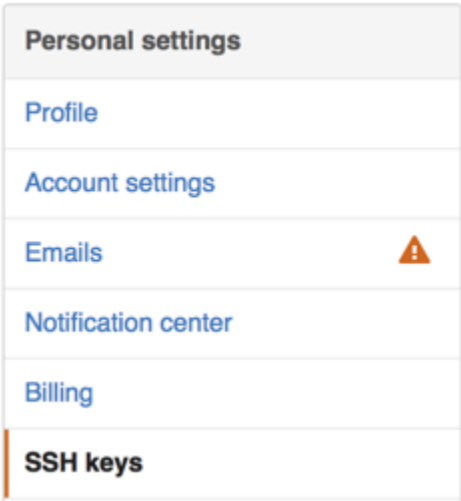
A screenshot of the GitHub sign-up interface. It features three input fields: the first contains 'greta-p' with a green checkmark; the second contains 'greta-p@duckietown.com' with a green checkmark; the third contains a masked password '.....' with a key icon and a green checkmark. Below the fields is a text requirement: 'Use at least one lowercase letter, one numeral, and seven characters.' A large green button labeled 'Sign up for GitHub' is positioned below the text. At the bottom, a paragraph states: 'By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy policy](#). We will send you account related emails occasionally.'

Verify email.

Go to settings.



16.2. Create an SSH Key on your machine



Need help? Check out our guide to [generating](#)

SSH keys

There are no SSH keys with access to your a

Generate the ssh key with the command:

```
$ ssh-keygen -h
```

The output will look something like this:

```
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa): /home/ubuntu/.ssh/your
name@robot name
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/your name@robot name
Your public key has been saved in /home/ubuntu/.ssh/your name@robot name.pub
The key fingerprint is:
4c:f8:c4:72:57:ba:ce:f1:e6:97:13:0d:27:39:43:ef your name@robot name
The key's randomart image is:
+--[ RSA 2048 ]-----+
|           .          |
|      o   o   .      |
|    o = o   . o      |
|   B . .   * o       |
|   S o   O           |
|      o o   . E      |
|      o o   o        |
|      o   +          |
|      . . .          |
+-----+

```

CHAPTER 17

SD Image to remote control

CHAPTER 18

RC remote control

CHAPTER 19

Joystick plus camera output

CHAPTER 20

Wheel calibration

CHAPTER 21

Camera calibration

CHAPTER 22

Taking a log

CHAPTER 23

Create a ROS package

CHAPTER 24

Reference for useful software commands

This is a reference for all commands mentioned in the documentation.
For each one, please add a description or a link to documentation/tutorials online.

24.1. SSH

(To write)

```
$ sudo apt-get install ssh
```

24.2. Byobu

(To write)

```
sudo apt-get install byobu
```

24.3. VIM

(To write)

```
sudo apt-get install vim
```

24.4. (htop)

Use htop to monitor CPU usage:

```
sudo apt-get install htop
```

24.5. Git and related commands

Additional utilities for git:

```
sudo apt-get install git-extras
```

(To write)

CHAPTER 25

Integrate package in the architecture

CHAPTER 26

LED Setup

CHAPTER 27

Acquiring the parts for Duckietown (2017)

CHAPTER 28

Traffic Light Assembly

PART 3

Modeling

TODO:

CHAPTER 29

Kinematics of Duckiebot

TODO:

PART 4
How to contribute

CHAPTER 30

Accounts

30.1. Complete list of accounts

Currently, Duckietown has the following accounts:

- Github: for source code, and (maybe) issue tracking;
- Slack: a forum for wide communication;
- Twist: to be used for instructors coordination;
- Google Drive: to be used for instructors coordination, maintaining TODOs, etc;
- Dropbox Folders (part of Andrea's personal accounts): to be abandoned;
- Vimeo, for storing the videos;
- The duckietown-teaching mailing list, for low-rate communication with instructors;
- We also have a list of addresses, of people signed up on the website, that we didn't use yet;
- The Facebook page.

We have to decide about issue tracking - (leaning towards Github).

30.2. For Fall 2017

As a student in Fall 2017, these are the accounts that you need:

- A Github account and membership of the Duckietown organization.
- A Slack account, for team discussion and organization.
- Dropbox: we will have 1 account that can be shared for uploading files (movies, logs). It is not necessary to install the Dropbox client on ones computer or to download all the files.

As an instructor/TA for the Fall 2017 class, in addition to the accounts above, these are the accounts that you need:

- Twist: for class organization;
- Google Docs, used to maintain TODOs.

30.3. For other contributors

If you are an international contributor:

- Sign up on Slack, to keep up with the project.
- (optional) Get Github permissions if you do frequent updates to the repositories.

CHAPTER 31

Contributing to the documentation

31.1. Where the documentation is

All the documentation is in the repository `duckietown/duckuments`.

The documentation is written as a series of small files in Markdown format.

It is then processed by a series of scripts to create this output:

- [a publication-quality PDF](#);
- [an online HTML version, split in multiple pages and with comments boxes](#).

31.2. Editing links

The simplest way to contribute to the documentation is to click any of the “✎” icons next to the headers.

They link to the “edit” page in Github. There, one can make and commit the edits in only a few seconds.

31.3. Comments

In the multiple-page version, each page also includes a comment box powered by a service called Disqus. This provides a way for people to write comments with a very low barrier. (We would periodically remove the comments.)

31.4. Installing dependencies for compiling the documentation

Let `DUCKUMENTS` be the base directory for the documentation.

Download the `duckuments` repo in that directory:

```
$ git clone git@github.com:duckietown/duckuments.git $DUCKUMENTS
```

Cd into directory:

```
$ cd $DUCKUMENTS
```

On Ubuntu 16.04, create a virtual environment using `virtualenv`:

```
$ virtualenv --system-site-packages deploy
```

In other distributions you might need to use `venv`:

```
$ venv deploy
```

Activate the virtual environment:

```
$ source $DUCKUMENTS/deploy/bin/activate
```

Install some dependencies:

```
$ sudo apt-get install libxml2-dev libxslt1-dev
$ sudo apt-get install libffi6 libffi-dev
$ sudo apt-get install python-dev python-numpy python-matplotlib
```

Clone the mcdp external repository:

```
$ cd $DUCKUMENTS
$ git clone -b duckuments git@github.com:AndreaCensi/mcdp.git
```

Install it and its dependencies:

```
$ cd $DUCKUMENTS/mcdp
$ python setup.py develop
```

Depending on your system, you might need to install these other dependencies: (It should not be necessary on Ubuntu 16 given the `apt-get` commands above.)

```
$ cd $DUCKUMENTS
$ pip install numpy matplotlib
```

Ensure the latest version (>6) of nodejs is installed.

Run:

```
$ nodejs --version
6.xx
```

If the version is 4 or less, remove nodejs:

```
$ sudo apt-get remove nodejs
```

Install nodejs using [the instructions at this page](#).

Next, install the necessary Javascript libraries using npm:

```
$ cd $DUCKUMENTS
$ npm install MathJax-node jsdom@9.3 less
```

Install PrinceXML from [this page](#).

TODO: instructions for fonts

31.5. Troubleshooting installation problems

1) Installing (nodejs) packages

The only pain point in the installation procedure has been the installation of nodejs packages using npm. For some reason, they cannot be installed globally (`npm install -g`).

Do not use `sudo` for installation. It will cause problems.

If you use `sudo`, you probably have to delete a bunch of directories, such as: `RBR00T/node_modules`, `/.npm`, and `/.node_modules`, if they exist.

31.6. Compiling the documentation

Make sure you have deployed and activated the virtual environment. Then:

```
$ cd $DUCKUMENTS
$ make duckuments-dist
```

This creates the directory `duckuments-dist`, which contains another checked out copy of the repository, but with the branch `gh-pages`, which is the branch that is published by Github using the “Github Pages” mechanism.

At this point, please make sure that you have these two `.git` folders:

```
$DUCKUMENTS/.git
$DUCKUMENTS/duckuments-dist/.git
```

To compile the docs, go in the `DUCKUMENTS` directory and run `make compile`:

```
$ cd $DUCKUMENTS
$ make all split
```

This creates the following files:

- `duckuments-dist/master/duckiebook.html` is a single-page HTML of everything.
- `duckuments-dist/master/duckiebook.pdf` is the PDF version.
- `duckuments-dist/master/duckiebook/index.html` is the first page of the version with each chapter on a different page.

31.7. Deploying the documentation

To deploy the documentation, jump into the `DUCKUMENTS/duckuments-dist` directory.

Run the command `git branch`. If the out does not say that you are on the branch `gh-pages`, then one of the steps before was done incorrectly.

```
$ cd $DUCKUMENTS/duckuments-dist
$ git branch
...
* gh-pages
...
```

Now, after triple checking that you are in the `gh-pages` branch, you can use `git status` to see the files that were added or modified, and simply use `git add`, `git commit` and `git push` to push the files to Github.

CHAPTER 32

Features of the documentation writing system

32.1. Embedded LaTeX

You can use ***LaTeX*** math, environment, and references. For example, take a look at

$$x^2 = \int_0^t f(\tau) d\tau$$

or refer to [Proposition 1](#).

Proposition 1. (Proposition example) This is an example proposition: $2x = x + x$.

The above was written as in [Figure 2](#).

```
You can use  $\LaTeX$  math, environment, and references.
For example, take a look at

\[
  x^2 = \int_0^t f(\tau) \text{d}\tau
\]

or refer to \[Proposition example\].

\begin{proposition}[Proposition example]\label{prop:example}
This is an example proposition:  $2x = x + x$ .
\end{proposition}
```

Figure 2. Use of LaTeX code.

TODO: other LaTeX features supported

32.2. Other interesting features

TODO: to write

32.3. Limitations

There are some limitations:

- Please use the string `&\$` to write the dollar symbol $\$$, otherwise it gets confused with LaTeX math materials. Also notice that you should probably use “USD” to refer to U.S. dollars

CHAPTER 33

Bibliography

- [1] [Jacopo Tani](#), [Liam Paull](#), [Maria Zuber](#), [Daniela Rus](#), [Jonathan How](#), [John Leonard](#), and Andrea Censi. **Duckietown: an innovative way to teach autonomy**. In *EduRobotics 2016*. Athens, Greece, December 2016.  [pdf](#)
- [2] [Liam Paull](#), [Jacopo Tani](#), Heejin Ahn, Javier Alonso-Mora, Luca Carlone, Michal Cap, Yu Fan Chen, Changhyun Choi, Jeff Dusek, Daniel Hoehener, Shih-Yuan Liu, Michael Novitzky, Igor Franzoni Okuyama, Jason Pazis, Guy Rosman, Valerio Varricchio, Hsueh-Cheng Wang, Dmitry Yershov, Hang Zhao, Michael Benjamin, [Christopher Carr](#), [Maria Zuber](#), [Sertac Karaman](#), [Emilio Frazzoli](#), [Domitilla Del Vecchio](#), [Daniela Rus](#), [Jonathan How](#), [John Leonard](#), and Andrea Censi. **Duckietown: an open, inexpensive and flexible platform for autonomy education and research**. In *IEEE International Conference on Robotics and Automation (ICRA)*. Singapore, May 2017.  [pdf](#)
- [3] Tosini, G., Ferguson, I., Tsubota, K. . Effects of blue light on the circadian system and eye physiology. *Molecular Vision*, 22, 61–72, 2016 ([online](#)).