

Lab 1: Setup and Lab Basics

Exercise 1. (*Setup*)

Download the [provided virtual machine](#) and follow its readme.md file to finalize the setup.

Make sure the tag '2122' is checked out in the CPM Lab software repo. `git status` should return 'HEAD detached at 2122'. If it does, you can skip the steps of building the software. If it does not, check out the tag with `git checkout 2122` and rebuild the software.



More information on the CPM Lab can be found in the [documentation](#).

Exercise 2. (*GitHub*)

If you are unfamiliar with git, go through the introduction to git provided through moodle.

Exercise 3. (*Repository*)

- a) Create a private GitHub repository for version control. For smooth execution of your code in the lab, clone your repo such that you end up with the following folder structure (example for GitHub repository TEAMREPO):

```
$HOME/dev
|-- software
    |-- cpm_lib
    |-- high_level_controller
        |-- examples
        |-- TEAMREPO
        |-- ...
    |-- indoor_positioning_system
    |-- ...
```

The path to your repository will be called TEAMREPO in the following. The path to the CPM Lab software will be called CPMLAB.

Initialize your repo with the code base given at [GitHub](#). Checkout the tag '2122'.

- b) If you are working in the CPM Lab: Navigate to your repo and configure your git username and email with

```
git config user.name "FIRST_NAME LAST_NAME"
git config user.email "MY_NAME@example.com"
```

Please do not use the option `--global` as you are sharing a user account with others.

Exercise 4. (*Controlling the vehicles*)

Open MATLAB. Read [how to run a high-level controller](#).

- a) Execute the example CPMLAB/high_level_controller/01_direct_control/main.m to see the vehicle moving in a circle.
- b) The file CPMLAB/cpm_lib/dds_idl/VehicleCommandDirect.idl defines the message format of messages sent to control the vehicle in direct control mode. What are the contents of the interface definition language (IDL)-file? How is the message sent in MATLAB?
- c) The file CPMLAB/cpm_lib/dds_idl/VehicleStateList.idl defines the message format of messages sent to give information about the vehicle states. What are the contents of the IDL-file? How is the message read in MATLAB?
- d) Repeat Task b) and Task a) for the path tracking mode.
- e) Repeat Task b) and Task a) for the trajectory following mode.

Checkpoint 1

Get a tutor to check your work. You should be able to

- run the path tracking example in your own setup
 - explain how communication between components is realized in the lab
 - explain the contents of a `VehicleStateList` message
 - show a GitHub-repository with the code template and all team members
 - explain the git commands `pull`, `add`, `commit`, `push`
-