Chair of Cyber-Physical Systems in Production Engineering
Department of Mechanical Engineering
Technical University of Munich

Walk through - “Keyboard control”

denis.hoornaert@tum.de

August 19, 2021

Objectives

In the present lab, you will be asked to complete few steps in order to integrate a keyboard controller to the project and control the car. More accurately, the students will add a provided node to the existing ROS project, start the simulator, see how the new node is integrated with the remaining of the environment and observe the communication between through the topic.

6.1

In the current project you have downloaded, you will find in `\node` the program called `keyboard.cpp`. As mentioned in *Lab 5: F110 ROS simulator*, while that program will be compiled thanks to `CMakeList.txt`, it will not be added to the project and launched alongside the simulator. Following the guidelines present in `\launch\simulator.launch`, add the keyboard node to the project.

> Remark:
>
> Make sure to respect the naming convention! As a reminder: you must use your name as a prefix for the package name, the topics' name and the nodes' name ?

6.2

Launch the ROS project in one terminal and on a second terminal, check using the `rqt_graph` (see *Lab 4: the ROS tools* for the complete command) command the state of the project:

- Has the `keyboard` node been added?
- To which node is it linked?
- Through which topic is the communication being performed?
- Does the topic name match with the content of `keyboard.cpp`?

6.3

First, using a third terminal, list the ROS topics being used and localise the one enabling the communication between the `keyboard` node and the simulator. Secondly, print on the third terminal the content of the identified topic. (Hint: look at subsection *Lab 4: the ROS tools*).

What is being printed on the terminal ?

Go back to the first terminal where you have launched the simulator. Thanks to this terminal and the `keyboard` node scanning the input, you should be able to control the car on the simulator using the `W`, `S`, `A` and `D` keys. What happens on the third terminal when you press the aforementioned keys and how does the car behave to these inputs?

> Note:
>
> The keyboard control is only possible by pressing the keys on the terminal. In other words, once you have launched your project, you must select the terminal from which you launched the simulator instead of the `rviz` window that just appeared.

6.4

Modify the existing `keyboard.cpp` in order to implement a manual braking functionality (i.e. the car stops) when the user press the space bar.