

ELE Prototyping SoSe 2021

Programming

If you have any questions during the week, send an email to
kristian.rother@hshl.de

ELE Prototyping Programming

Task 1 - Overview

- **Deadline: 3rd of June 2021, 20:00h**
 - Send your file **via email** to kristian.rother@hshl.de (subject: ELE Prototyping Task 1)
 - **AND uploaded the file to your GitHub account**
- **Presentation of results: 4th of June 2021 (WebEx, see schedule)**
- **Code in C (not C++)**
- Feel free to use any programming environment you like. If you're unsure, you can use Visual Studio Code
 - <https://code.visualstudio.com/>
 - Install the extension for C/C++
 - Optionally install the Code Runner extension
 - Install/setup the compiler for your platform
 - Windows: <https://www.javatpoint.com/how-to-run-a-c-program-in-visual-studio-code>
 - Mac: <https://code.visualstudio.com/docs/cpp/config-clang-mac>

ELE Prototyping Programming

Task 1 - Description

- Your robot gets a **representation of the world** it is navigating in
- This representation is a **character array in C**
 - Empty space: O
 - Wall: #
 - Target: T
 - Your robot: R
- Based on this representation, the robot should be able to move north, south, east and west
- If the robot reaches the target (T), your robot succeeds
- If the robot crashes into a wall (#) or the maximum number of steps (200) is exceeded, your robot fails

ELE Prototyping Programming

Task 1 – Your Task 1/3

- Download the file `robot_world_1.c`. It contains the simulation of the rescue scenario
- **Set up your development environment and compile and run the simulation** (`robot_world_1.c`), the dummy robot will crash into a wall
- **Implement** the function ***`int move(char *world)`***. Replace the existing dummy code with your own code
- The function gets a character array that represents the world. Parse this array and use it to decide what your robot should do. Each element of the array can have one of the following values
 - 'R' = robot position
 - 'O' = Empty space (it's the uppercase letter O, not a zero)
 - '#' = Wall
 - 'T' = target to rescue
 - '\n' = Newline. Only used to format the map. Do not use this value.
- Your function should return an integer that represents the direction you are moving in
 - North: 1
 - East: 2
 - South: 3
 - West: 4
- **Do not change anything except the ***`int move(char *world)`*** function**

ELE Prototyping Programming

Task 1 – Your Task 2/3

- You should **implement the function *int move(char *world)*** so that your robot is able to navigate to the target without hitting a wall
- If you hit a wall, the simulation will end

ELE Prototyping Programming

Task 1 – Your Task 3/3

- **Send** your final `robot_world_1.c` file to kristian.rother@hshl.de **via email AND upload the code to your GitHub** before the deadline
- **Prepare a presentation for Friday** to explain your code
- Think about **what should change in the simulation to make it more realistic** (we will discuss this on Friday)
- Important note: I will copy your *int move(char *world)* function into my test environment. The environment contains different maps, not just the one provided to you. **Do not hard code a solution.** The robot should handle different maps
- All maps contain only walls and the target