COMP105P Lab Tasks 1

Tasks

The tasks of the first practical are purposely straightforward to complete. The practical itself is mainly dedicated to get familiar with the use of the software.

Preparation: Read the document "*SimpleIDE Libraries - Student report*". It is available on moodle. It has been compiled by a Nuffield foundation placement student. In addition to the official ActivityBot documentation, the current report provides a bunch of useful code samples and video demos. All written from a beginner's point of view.

Take the time to experiment the different functions/libraries available so you are ready to complete more complex tasks in the coming weeks.

Once done, please complete and demo the following tasks.

- Task 1.0. Get the robot to draw a 1-metre square using the motor encoders to measure the length of the sides and the angles turned.
- Task 1.1. Get the robot to go spin in a circle on the spot.
- Task 1.2. The teaching assistant will give you a random sequence of letters containing any of the following A,E,F,H,I,K,L. Get the robot to write that word on the floor. You can't hard code words. You will need to implement a function that takes as input a string and then get the robot to draw it.

Hints

Here are some hints to get you going.

- You need to install the robot simulator on your laptop. See guidelines on moodle. The documentation of the robot simulator is included in the zip file.
- Put each task (or subtask) in its own self-contained C file and name it appropriately. You will end up with three files (one for each task), each of which should compile and execute cleanly when ran in the simulator. When submitting bundle all the tasks in a .zip and name it appropriately. There are a lot of students in the class and we do not want to have to guess which group originated which file and which zip archive - please take care to adhere to a sensible naming convention. You can use the following naming for your zip file *groupNumber_taskNumber.zip*, for example ***group25_task1.zip***.
- Keep your code neat and tidy. Make sure you do everything as obviously as possible. Please add comments where it may not be immediately apparent what you are trying to do. We will actually be looking at the code you submit, so do not produce a jumbled mess.

- Make things as modular as possible - as mentioned in the first lecture your later tasks   will build on earlier ones and the level of complexity will increase gradually. This usually means you will have to do the same basic things over and over again (e.g. parsing the readings of a certain type of sensor) - why not put those things in separate functions (and files later on when the tasks get big) that you can reuse.

Type of assessment: formative. Refer to lecture 1 for explanations

Marking: Each part is binary marked

- 50% if the demo of all parts of the task sheet is successful
- 50% for a working submitted code

Deadline: Deadlines are always specified in the submission link on moodle.

The teaching assistants are:

Group A
Stanislav Kikot, kikot@dcs.bbk.ac.uk

Group B
Temitope, Odedeyi, temitope.odedeyi.13@ucl.ac.uk
Mohammad Hamid, mohammad.hamid.16@ucl.ac.uk

Group C
Roxana, Ramirez Herrera, roxana.herrera.15@ucl.ac.uk
Manni Singh, manni@dcs.bbk.ac.uk

Group D
Butters, Daniel, daniel.butters.16@ucl.ac.uk (Lead TA)
Magbagbeola, Morenike, morenike.magbagbeola.16@ucl.ac.uk

Group E
Khan, Muhammad Azeem, m_azeemk@yahoo.com
Kabir, Mahmoud Tukur, kabir.tukur.15@ucl.ac.uk

Note: If you have any questions about the assignments feel free to ask your teaching assistants. Please note that we will not do the tasks for you or code anything for you.