CSCE 274 – Robotics Design and Applications – Fall 2019

Project 1 (10% over the final grade)

Assigned: September 10, 2019
Due: October 03, 2019

The purpose of this assignment is to make sure you can write and execute programs for our iRobot Create 2 robots.

Instructions

Please read carefully the following tasks and program the robot accordingly. *You should do this assignment in the groups that have been assigned.*

The instructions on how to use the robot and how to submit the project assignment is on the webpage of the class, at the following links

https://drive.google.com/file/d/1Z4sSVl08r5S98r2kKA0xmwTWbPnh8d63/view

https://drive.google.com/file/d/1JRj0926o3-jKDnJP5l_T_xM4pE0sSSgy/view

Tasks

Task 1. Write an *interface*[1] for the serial communication, that includes:
   a. Connection to the serial interface.
   b. Sending of commands.
   c. Reading of data.
   d. Close the connection.

Task 2. Using the interface in Task 1, write an *interface* for the robot that:
   a. Control the state of the robot (Start, Reset, Stop, Passive, Safe).
   b. Read the state of the buttons.
   c. Send a Drive command to set the velocity and the radius of the wheels, given the two as arguments.

Task 3. Write a *program* that utilizes the previous interfaces and:
   a. Initializes the robot, by setting it in *passive* and *safe* mode.
   b. If the robot is stopped, once the *clean/power* button is pressed, given an input **N**, move counterclockwise along a **regular polygon** with **N** sides and total perimeter of **2m** (meters). The robot stops once the polygon is covered. Your solution should work for any valid value of **N**.
   c. If the robot is moving, when the clean/power button is pressed, stop the robot when it reaches the current goal vertex.
   d. Note that the program should continue listening for button presses, both in (b) and (c) cases, until the program is terminated.

---

1 *Interface* here refers to the general concept of Application Programming Interface (API), which is a set of programming instructions (that can be in classes or methods) that can be used to build your own application – i.e., that in Task 3. An example in Python that uses classes can be found in https://python.swaroopch.com/oop.html.

Comments

Your programs will be evaluated based on both their functionality and their coding style. In the notes for iRobot Create 2 you can find an informal style guide to help give you an idea of what is expected together with the coding style that you should follow. Also, as for Task 1 and Task 2 you are providing interfaces, it is recommended to use one or more separate files from the one of Task 3.

The (b) objective of Task 3, might seem confusing at first glance but there is a straightforward solution. You can use two very simple formulas that are applied to all regular polygons for calculating for any given **N** and any perimeter **P**, the angles and the length of every regular **N**-gon assuming N>2 and P>0.

Although completing the requirements above is enough to obtain full credit, I also encourage you to explore and experiment with the robots. What other sensors are available? What other actions can the robot execute? This will help for the next project assignments.

Evaluation

The solution will be evaluated according to the following.

If the robot does not move there would be -60% penalty.

Task 1. functionality (15):

- Connection to serial interface.
- Proper sending of commands (with correct encoding).
- Proper reading of data, returning raw data.
- Close the connection.

Task 2. functionality (15):

- Control the state of the robot.
- Proper reading of the state of each button.
- Proper sending of Drive commands.

Task 3. functionality (30):

- Proper initialization of the robot.
- Accurate regular N-gon shape.
- Proper reading of the button state even when the robot is moving.

Style (20):

- No duplication of executable code?
- No magic numbers?
- Names match functionality?
- Adequate comments?
- Comments match code?
- Consistent formatting?

Documentation (20):

- Report is complete and clear?
- Required sections exist?
- Free of typos and grammatical errors?