CPS 499/592 — *Intro to Robotics*
Spring 2023 — *Lab* 04

Assigned: 2023-03-01
Due: 2022-03-22

Overview

The purpose of this assignment is two-fold:

1. To ensure that you are comfortable using the software suite introduced last lab to program the iRobot Create2.
   - The code can be found at `https://github.com/DrR0b0tN1ck/Create2_Library`
2. To code more "advanced" functionality than what has been asked of you up to this point.

Task 1: Recover Lab 02 functionality

This task requires you to re-implement the functionality from Lab 02 using the "new" software suite.

Specifically,

1. Read specific sensors
   - Read the robots wall signal and its four cliff sensor signal values. (Each of these should be an unsigned 16-bit integer.)
   - Read all of the sensors in **Group Packet ID # 3** (Packets 21-26).
   - Display those sensor values in a human-readable format, identified with English text strings that contain
     - what the packet corresponds to,
     - the value returned by the robot,
     - any applicable units.

     in the Tkinter window.

     Example: `Battery Charge:  60531 mAh`
2. Control the LED's.
   Specifically, create a periodic timer that will toggle the robot's LEDs such that **When activated** … which also means this functionality can be paused …, the robot will **alternate/toggle** between:
   - Turning on the spot/dock LEDs **and** turning off the power led, check robot, debris LEDs
   - Turning off the spot/dock LEDs and turning the power led "RED" at full intensity and turning the check robot and debris LEDs on.

Task 2: Driving

This task requires that you implement two different versions of “safe” driving. The difference being the sensor we are reliant on to *detect* a bump.

1. The first version should use the **Bumps and Wheel Drops** in `Packet ID: 7`. If the robot has it’s wheeldrop sensors go off, or detects a bump, it should stop.
2. The second version should use the **Light Bumper**. You can either use `Packet ID: 45` or the higher resolution individual versions (IDs 46-51).

The idea is that when you press a specified key, the robot will start driving and will stop when it detects a wall based on the aforementioned sensors. So you should have two buttons, one for each mode.

Task 3: Distance Driving

This task asks you create a new method called `driveDistance` to complement the other drive commands in `create_robot.py`:

```
def driveDistance(self, velocity, distance):
""" Any useful comments explaining the method """
```

The intuition is that this command will drive the robot forward at the specified velocity for the specified distance. The two stopping conditions are:

1. Encountered an obstacle prior to traversing the specified distance.
2. Did not encounter any obstacles.

Both options should print out how far the robot travelled.

You are given full autonomy in how you decide to tackle this but your design choices should appear in the report, which **will** naturally be longer because of the third task.

CPS 499/592 – Deliverable – Lab 04

I will be accepting one document for the deliverable. Below are the details regarding this document.

Filetype: pdf if it is not a pdf ... 5 points will be deducted
Filename: Lab 04.pdf
The header of the document should look like the following example:

Team Members: Nicholas Stiffler, Student1 (YYYY), Student2 (ZZZZ)
Course: CPS 499/592
Assignment: Lab 04

There are 2 major components to the deliverable:

1. **Python Code**
   All of the code that you used needs to be included in the submission in a folder called "code".

2. **Report**
   This component requires you do the following:

   - Write a brief report about the project describing the steps your team took to accomplish each task. Assume each task will require a *minimum* of a half-page of explanation in the report.
     - What challenges did you face?
     - What did not work as you had anticipated?
     - What did your team do to overcome these challenges?
     - If you had more time what would you do differently to improve the code?
     - etc.

Below is the grading rubric I will use when evaluating your submission.

- The deliverable should be an archive (.zip, .tgz., etc) that contains the following
  - A pdf file for the report.
  - A "code" directory that contains all of the code you used for the assignment.
- The demo requires a time, scheduled either outside of class or on a designated lab day where you will show Dr. Stiffler that your project works, and he will ask questions about your design decisions, etc.
- **Your submission will not be graded without demoing the project**

CPS 499/592 – Cover Sheet – Lab 04

Team number: ______
Names:

____

____

____

____

Task 1 functionality (20):

- □ Key(s) for both operations appears in the *help text*
- □ Reading specific sensors component
- □ Toggling LEDs component

Task 2 functionality (20):

- □ Key(s) for both drive safe modes appear in the *help text*
- □ Drive Safe using **Bumps and Wheeldrops**.
- □ Drive Safe using **Light Bumpers**.

Task 3 functionality (20):

- □ Write a function that *minimally* accepts the distance as a parameter. Velocity may be hard-coded.
- □ Robot drive's safely (ie stop if an obstacle is encountered).
- □ Robot drive's the specified distance.
- □ Output the distance travelled.

Style (20):
The following refers to your code, which should be submitted **as part of your Team's deliverable document**.

- □ One function per command?
- □ 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? (report and code)

Other comments:

Total: