Escola Superior de Tecnologia e Gestão Instituto Politécnico da Guarda	**Laboratorial Work**

Nº 5

Group: 3.1 ←Identify your group

Number	1012164	Name	Carina Tomé
Number	1012208	Name	Dário Ribeiro
Number	1012180	Name	Pedro Sanches
Number	1012396	Name	Rui Manta

Fill in the header record with the names and numbers of the group members and add "(missing)" next to the name, in case any member of the group missed the class. After finishing the laboratorial work, create a PDF doc called "Grupo X.X – LW X.PDF" submit it via the form at http://bit.ly/2TL0IKS.

If the answers to the questions is Java code, you must use color black and font Courier New to write the code. Also, indent all the code. A penalty of 50% will be applied if you forget to do this.

1. Complete the following text about the bumper sensors used in your robot.

   The RB kit has two bumper sensor that can be used to detect obstacles in complement to the sonars because they have blind spots. The bumper sensors are simple press buttons that can be connected between the ground pin and the signal pin of a digital port, and it is necessary to activate the pullup resistor in the correspondent port, so that the bumpers work correctly. Considering the electronic circuit studied in class for the connection of a bumper to the port, the values obtained by reading the port are 0 and 1, respectively if the bumper is active (colliding), and inactive (not colliding).

2. Complete the following text about the "white line" sensor in your robot.

   The RB kit has a "white line" sensor that can be used to detect the white lines that signal the entrances of the rooms of the arena. This sensor has an interface of type analog output and must be connect to one of the analog ports between A4 and A7. The way the sensor works is by emitting a beam of infrared light and then detecting the reflection. The sensor is mounted a short distance from a surface and its output can represent how reflective it is the surface. If the surface is very reflective (such as the white line) the sensor output that we can read in the port is around 0 and if the surface is just slightly reflective (blue / black carpets), the sensor output is around 1023.

3. Develop a new version of the navigateRightState() developed in last class that allows the robot to do the following two thigs. **The answer to this question is only valid if you demonstrate it to the professor, during this class.**

1º While navigating by following the right wall, if a collision is detected by one of the bumpers, the robot does a preprogramed maneuver to recover from that collision. The maneuvers can be a simple backward movement as illustrated in the following figure.
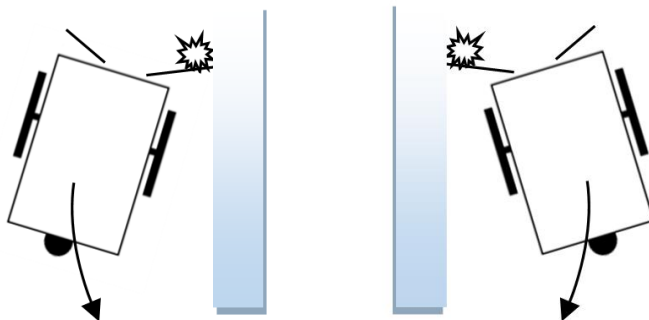
2º While navigating by following the right wall, the robot counts and displays in the LCD the white lines detected. In order to avoid multiple detections and counts of the same white line, you should implement some algorithm like the following.

```
. . .
value = read the value of the line sensor;
  if(value < threshold) { //Is a white line.
    move forward during a very brief moment just to pass by the
white line;
    totalLines = totalLines + 1;
    display totalLines;
  }
. . .
```

Declaration of the hardware objects used by the function:

```
private static IntelliBrainDigitalIO leftBumper;
private static IntelliBrainDigitalIO rightBumper;
```

Creation of the hardware objects used by the function:

```
leftBumper = IntelliBrain.getDigitalIO(1);
leftBumper.setPullUp(true);
rightBumper = IntelliBrain.getDigitalIO(10);
rightBumper.setPullUp(true);
```

New version of the behavior:

```
private static void navigateRightState() throws InterruptedException
{
        if(rightDist <= 0)
                rightDist = 100.0f;
        if (!leftBumper.isSet()) {
                move(-7,4);
                Thread.sleep(1000);
        } else if (!rightBumper.isSet()) {
                move(-7,-2);
                Thread.sleep(1000);
        }
        if (fairchild.sample() < 70) {
                move(8,0);
                Thread.sleep(250);
                lines++;
        }
        lcd.print(1, "Lines Passed: "+lines);

        if (frontDist < 20.0f) {
                leftMotor.setPower(-8);
                rightMotor.setPower(8);
                Thread.sleep(450);
        }
        float error = rightDist - MIN_DISTANCE_RIGHT;
```

```
int delta = (int) (error * GAIN);

if (delta > 5)
        delta = 5;
else if (delta < -5)
        delta = -5;
move(BASE_POWER, delta);
}
```