



Escola Superior de Tecnologia e Gestão  
Instituto Politécnico da Guarda

## Laboratorial Work

### Nº 8

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 (Ausente)

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/2TLOIKS>.

If the answers to the questions is Java code, you must use color black and font Courier New, size 10, to write the code. Also, indent all the code. A penalty of 50% will be applied if you forget to do this.

The answer to this question is only valid if you demonstrate it to the professor, during this class.

1. The robot must be able to start the run at the standard starting location, positioned parallel to any of the two walls that form the corner at that location.

Tip: Add the code required for the robot to sense its orientation based on the distances measured by the SONAR sensors and correct this orientation if necessary before starting navigation.

Declaration of new constants, if any

Declaration of new objects and variables, if any

```
private static RangeFinder mLeftSonar;
```

Creation and initialization of objects and variables, if any

```
mLeftSonar = new ParallaxPing(IntelliBrain.getDigitalIO(3));
```

New functions, if any

Call of the new functions, if any



Escola Superior de Tecnologia e Gestão  
Instituto Politécnico da Guarda

## Laboratorial Work

### New code, if any

```
Dentro do waitstate adicionamos a condição if
    (getDistance(mLeftSonar) < MIN_DISTANCE_LEFT) {
        rotateAngle(-90);
    }
```

2. Use the UVTron sensor to improve the navigation of the robot while looking for the flame. At the entrance of each room the robot uses the UVTron sensor to detect the presence of the flame in the room. If the flame is present, the robot enters the room and looks for the flame. If not, the robot makes a maneuver and navigates toward the next room. The UVTron sensor is connected to an analog port, so it works as the line sensor. If no flame is present, the sensor returns 0 and if the flame is present it returns a value different from 0.

Don't answer by presenting all your program. Use the following sections to present your new code. Delete what doesn't apply.

### Declaration of new constants, if any

### Declaration of new objects and variables, if any

### Creation and initialization of objects and variables, if any

### New functions, if any

### Call of the new functions, if any

### New code, if any

```
//Dentro do Navigate_Right state

int tag = getFloorTag();
    if (tag == LINE_TAG) {
        stop();
    }
```



```
if (mUVTron.sample() > 0) {  
    mFlameLED.set();  
    return CENTER;  
} else {  
    move(-BASE_POWER, 0);  
    wait(1000);  
    rotateAngle(90);  
    return NAVIGATE_RIGHT;  
}  
}else if(tag == CIRCLE_TAG)  
    if (mFlame == true)  
        maneuverToReturnBack();  
    else  
        return WAIT;
```

3. Make sure your robot only senses the flame if it is inside a room so as to prevent false flames while navigating the corridors (due to the reflection of the sun on the walls, for example).

Don't answer by presenting all your program. Use the following sections to present your new code. Delete what doesn't apply.

Declaration of new constants, if any

Declaration of new objects and variables, if any

Creation and initialization of objects and variables, if any

New functions, if any

Call of the new functions, if any

New code, if any

// Mesma coisa que o Código anterior só que com a adição de uma condição para a Terceira e quarta  
//linha



```
if (tag == LINE_TAG) {
    mTotalLines++;
    mLcd.print(1, "L: " + mTotalLines);
    stop();

    if (mUVTron.sample() > 0) {
        mFlameLED.set();
        return CENTER;
    } else if(mTotalLines == 3 || mTotalLines == 4){
        return NAVIGATE_RIGHT;
    }else {
        move(-BASE_POWER, 0);
        wait(1200);
        rotateAngle(100);
        return NAVIGATE_RIGHT;
    }
} else if(tag == CIRCLE_TAG)
    if (mFlame == true)
        maneuverToReturnBack();
    else
        return WAIT;
```