



Escola Superior de Tecnologia e Gestão
Instituto Politécnico da Guarda

Laboratorial Work

Nº 2

Group: 31

Name	Edinilson Wagner	Nº	1011276
Name	Janilta Pires	Nº	1012063
Name	Janilza Simão	Nº	1012414

Fill in the header record with the name and number of the group members and add “(missing)” in case any member of the group has missed the class. After finishing the laboratorial work, add it to the group files section in Blackboard.

1. Identify and describe the 3 phases of the development of a program for the IntelliBriar-Bot.

Phase 1: Edit – Alterar o código

Phase 2: Build – Compila, liga e carrega o programa no robô

Phase 3: Test – Execução do programa

2. Describe what the Linker does.

R: Vai as classes declaradas no programa e cria o ficheiro binário (executável)

3. Relate the definitions in the left table with the file types in the right table. Use the letters.

A	MyBot.java
B	MyBot.class
C	MyBot\$.hex
D	MyBot.rjp

B	File generated by the Compiler.
D	File with the properties of the project.
A	File with the source code.
C	Executable file generated by the Linker.



4. Modify the program you did in class to introduce a new compilation error. Write down the error message you receive. Describe how the message relates to the actual error in your program.

R: Compiling...Found 1 syntax error in

"C:/Users/Janilza/Documents/IPG/2/2Semestre/Robotica/MyBot/MyBot.java":

```
9.    System.out.println("STOP pressed!")
```

```
      ^_____^
```

*** Syntax: ";" inserted to complete ExpressionStatement

Compilation failed

A mensagem de erro indica:

- Em que ficheiro estão os erros
- Em que linha
- ^ que indica exatamente em que parte da linha está o erro
- É uma possível solução para o erro encontrado

5. Look for information about the Display class on the API JavaDoc. Modify the program you did in class, to use the Display class to write on the LCD, instead of using the System.out.println instruction.

R:

```
import com.ridgesoft.io.Display;

public class MyBot {

    public static void main(String args[]) {

        Display display = IntelliBrain.getLcdDisplay();

        display.print(0,"Grupo 31");

    }

}
```



6. Try some of the examples that are in the folder C:\Programas\RoboJDE\Examples\IntelliBrain. They are examples about how to program the IntelliBrain hardware resources.

Try the examples about using:

- LEDs
- BUZZER
- Servo ports

Analog ports (make changes needed to use the analog sensor connected to port 7 in your robot). R: `private static final int PORT_NUMBER = 7;`

Analyze these examples and notice the pattern used to declare the objects to program the IntelliBrain hardware resources. In order to program an IntelliBrain hardware resource, we need to declare an object of the correspondent class, and we need to create the object using the correspondent methods of the IntelliBrain class.

Write the source code needed to declare objects for the following hardware resources.

LED: `LED statusLED = IntelliBrain.getStatusLed();`

BUZZER: `Speaker buzzer = IntelliBrain.getBuzzer();`

Motor port: `Motor motor = IntelliBrain.getMotor(1);`

Digital port: `IntelliBrainDigitalIO input = IntelliBrain.getDigitalIO(PORT_NUMBER);`

7. Open the project SimpleBot (C:\Program Files\RoboJDE\Examples\IntelliBrainBot\SimpleBot). Analyze the source code and execute it to see how the robot performs. Change the source code to make the robot move in circles instead of following straight lines. (Tip: change the powers applied to the motors and the time values that control the movement).

R: `//o motor gira sobre si próprio`

```
public void forward() {
```

```
    mLeftMotor.setPower(Motor.MAX_FORWARD);
```

```
    mRightMotor.setPower(0);
```

```
}
```

```
public void backward() {
```

```
    this.mLeftMotor.setPower(0);
```

```
    this.mRightMotor.setPower(Motor.MAX_REVERSE); }
```