

Using the Lego EV3 Robots

Dr Siobhán North

Department of Computer Science, The University of Sheffield

Building your robot

Each robot comes with a box of Lego Lego[®] components that can be made into a working robot. The main elements are:

- A named EV3 micro-computer that controls the rest of the components;
- Two large motors;
- A medium motor;
- An ultrasonic (distance) sensor;
- Two touch sensors;
- A colour sensor;
- A gyroscopic sensor;
- Seven cables;
- Wheels, gears, axles and assorted building blocks.

Motors are driven by motor ports which are labelled A, B, C and D. To drive a motor, you must connect its input port to one of the motor ports using a black cable. Sensors are connected to sensor ports which are labelled 1, 2, 3 and 4. Again, connect a sensor to an appropriate port using a black cable.

Your robot kit is initially configured as a three-wheeled “tribot”. You can add as many components as you like but try and keep this underlying configuration because you will have to return it to that structure at the end of the practical class.

The first thing you should do is turn it on by pressing the dark-grey button on the front of the EV3 micro-computer until the light comes on. When it is ready to use a menu will appear on the robot’s screen.

Programming the robot

The general principle is that a Java program running on a desktop computer will act as a “remote control” for the robot. The computer communicates with the robot via a wireless connection (not Eduroam).

First write your program. For this you will need the **ShefRobot** package. Extract the contents of *ShefRobot.zip* from the web page where you got these instructions into your working directory.

First you should import the `ShefRobot` package

```
import ShefRobot.*;
```

at the start of the program. In your Java program the class `Robot`, imported from `ShefRobot` represents your robot so it is very important not to call your Java program `Robot.java`.

In the main method you should first create a robot object which identifies your robot something like this.

```
Robot myRobot = new Robot("dia-lego-x9");
```

but you should substitute the name of your robot for the characters “x9” above. Your robot’s name is on a yellow sticker on the robot and is repeated at the top of the screen on the EV3 micro-computer. It will be a single capital letter followed by a single digit. In the statement above use the name in lower case.

Before you compile your program for the first time, in the command window (the one with the black screen) type

```
u:\myJava>cd ShefRobot
u:\myJava>setupconsole
u:\myJava>cd..
```

You only need to do this once per session unless you close your command window. Now you can compile and recompile the Java program in the normal way.

Controlling the motors

In the Java program the classes `LargeMotor` and `MediumMotor` represent each motor.

First give them names. This isn’t essential it just makes your program far more readable.

```
Motor leftMotor =
    myRobot.getLargeMotor(Motor.Port.B);
Motor rightMotor =
    myRobot.getLargeMotor(Motor.Port.C);
```

Always assuming your left motor is plugged into port B and the right one into C.

Now you can control them with statements like these:

```
myMotor.forward();    //Move forward indefinitely
myMotor.backward();    //Move backward indefinitely
myMotor.setSpeed(500); //Set the speed to 500
myMotor.stop();        //Stop myMotor completely.
```

The speed is in degrees per second and the range is 0-900. To turn the robot either stop one wheel while the other goes forward or make one go forward whilst the other goes backwards. To make the robot continue its behaviour for a specific length of time use

```
myRobot.sleep(1000); //Wait for 1 second.
```

Making a sound

The robot is able to make simple sounds through an internal speaker. This can be useful for testing sensors (e.g., make the robot beep if a touch sensor is pressed see below). Use statements like:

```
Speaker mySpeaker = myRobot.getSpeaker();
mySpeaker.playTone(500,200);
```

The parameters of playTone are first the pitch in Hz then the duration in milliseconds.

Using the sensors

The box of Lego that accompanies the robot has a variety of sensors which can be attached to the robot and connected to one of the sensor ports and then controlled by your program. For example to use a touch sensor first create an object to represent it. This depends on the port that the sensor is attached to. If your touch sensor is attached to port one (`Sensor.Port.S1`) then you create an object `myTouch` to represent the sensor as follows:

```
TouchSensor myTouch =
    myRobot.getTouchSensor(Sensor.Port.S1);
```

The other ports are S2, S3 and S4 and can be used in the same way. Now you can check if the sensor has been pressed by using statements like these:

```
if ( myTouch.isTouched() )
    mySpeaker.playTone(1000,200);
```

The robot will emit a short beep every time the touch sensor is pressed. The documentation for all the various sensors is available from the web page and at <http://ramsay-t.github.io/ShefRobot/>

The last statement your program executes should be the Robot's `close()` method, to disconnect cleanly.

```
myRobot.close();
```

If you have any problems you can test everything is working OK by running the `RobotSample.java` program available from the robot web page. It makes the robot shuffle forward a bit, beep and then shuffle back to its starting point. If that doesn't work it is likely to be a problem with the robot or its connection but if it does the problem is going to be in your program.

At the end of the session

Turn the robot off by pressing the light-grey button, navigating to the Yes option using the left and right buttons, and confirming with the central dark-grey button. Remove any sensors etc. and put all the Lego back in its box. Return the robot and its box of accessories to one of the technical staff.