Lego Rovers: Parent/Teacher Handbook and Trouble-shooting Guide

Louise A. Dennis

July 25, 2016

Contents

Chapter 1

Introduction

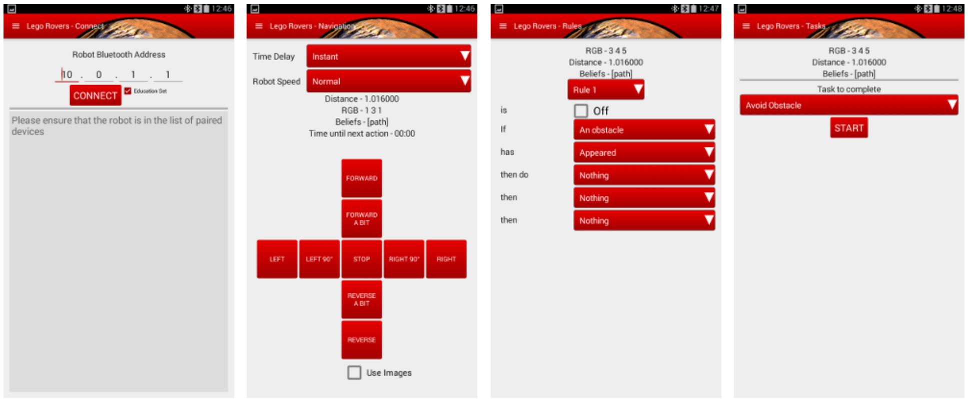
Lego Rovers is an activity aimed at children in Key Stages 2 and 3. It aims to introduce them to ideas from artificial intelligence, robotics and space science. The activity takes 30 minute to an hour and uses a LEGO Mindstorms EV3 robot and an app running on an Android tablet.

The Lego Rovers app and supporting material are free and can be downloaded from the Lego Rovers website (`http://legorovers.csc.liv.ac.uk`). Supporting material includes build instructions for the LEGO robot used in the activity, a workbook that can take a child through the activity, and two printable mats that can be used with the robot's RGB (Red, Green, Blue) colour sensor. The code that runs on the robot during the activity must be copied to a micro SD card which is placed in the robot's SD slot during the activity. Once the card is removed then the robot will revert to the Mindstorms framework that it comes with. The necessary files for the SD card are available from the downloads section of the Lego Rovers website. It is also possible to order pre-prepared cards from the website to be sent to UK addresses.

The app consists of several panes of which the main ones are shown in figure 1.1. The panes can be navigated between using a menu located by tapping in the top left hand corner of the app.

Connect Pane The Connect Pane is open when the app is started and allows the app to be connected to the EV3 robot.

Navigation Pane The Navigation Pane allows teleoperation (remote control) of the robot, both instantaneously and at the time delays associated with communication between the Earth, the Moon and Mars.



Lego Rovers - Connect

Robot Bluetooth Address: 10 . 0 . 1 . 1

☑ Education Set

Please ensure that the robot is in the list of paired devices

Lego Rovers - Navigation

Time Delay: Instant

Robot Speed: Normal

Distance - 1.016000
RGB - 1 3 1
Beliefs - [path]
Time until next action - 00:00

□ Use Images

Lego Rovers - Rules

RGB - 3 4 5
Distance - 1.016000
Beliefs - [path]

Rule 1

is □ Off
If: An obstacle
has: Appeared
then do: Nothing
then: Nothing
then: Nothing

Lego Rovers - Tasks

RGB - 3 4 5
Distance - 1.016000
Beliefs - [path]

Task to complete: Avoid Obstacle

Figure 1.1: The Lego Rovers app Panes

Rules Pane The Rules Pane allows the robot to be programmed using simple rules.

Tasks Pane The Tasks Pane contains three pre-programmed behaviours which demonstrate the use of the robot's sensors.

This is the parent teacher handbook for the Lego Rovers activity. It includes instructions for installing the app and setting up the robot for the activity (Chapter 2), example lesson plans (Chapter 3), more detailed explanations of the concepts introduced in the workbook and sample answers to the projects and challenges in the workbook (Chapter 4), links to additional resources for further exploration (Chapter 5) and a troubleshooting guide (Chapter 6).

The Lego Rovers activity and application are the result of several years of effort at the University of Liverpool. It has been generously funded both by the University and by the Science and Technology Facilities Council (STFC). It has also been supported by STEMNeT via the STEM Ambassador Scheme, and NASA and the UK Met Office via the NASA SpaceApps Hackathons. Individual thanks are owed to Joe Collenette (app programming), Matoula Kotsialou (build instructions and workbook), Katie Atkinson, Elisa Cucco, Maryam Kamali, Gillian Murray, Michael Fisher and Andy Wood.

Chapter 2

Installation and Set Up

This chapter discusses installing and configuring the LEGO Rovers activity. To run the Lego Rover activity you will need:

- An Android device running Android 4.4 (KitKat) or higher with a screen size of at least 7 inches. You can find your Android version by going to Settings on your device. **Important:** Your device must also be capable of "Bluetooth Tethering" – that is using an external paired Bluetooth Device for Internet access. Only some Android devices allow this (for instance the Google Nexus 7 with Android 6 does not).

- A LEGO Mindstorms EV3 set. The activity will work for both the Home Set and the Education Set but some of the usage instructions vary between the sets.

- An SD card with an image of the leJOS operating system. You can purchase and format your own SD Card or request a pre-made SD card from us for free (UK only) via the Lego Rovers website.

This chapter covers getting an SD card (section 2.1), installing the card on the EV3 brick (section 2.2), installing the application on the Android device (section 2.4), building the physical robot from LEGO (section 2.5), calibrating the robot sensors (section 2.6) and running the activity (section 2.7).

2.1 How to get an SD Card with leJOS

leJOS is a free open-source Java-based operating system for LEGO robots available from `http://www.lejos.org`. This is the basis for the program that runs on the EV3 robot during the Lego Rovers activity. EV3s can run from an SD card, so the activity can be installed simply by placing an SD card in the robot's SD slot and then removing it once the activity is over.

We supply SD cards with the leJOS operating system free to UK users. These may be ordered using a form on the Lego Rovers website. Alternatively you may create your own following the instructions below.

2.1.1 What You Need

To create an SD card containing LeJOS you will need the following items:

- A micro SD Card which is a least 2GB and at most 32GB

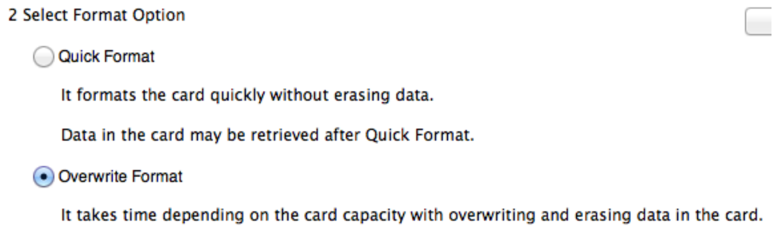
- A zip file containing the leJOS files available from the Downloads and Resources section of the Lego Rovers website.

2.1.2 Formatting the SD Card

The micro SD Card must be FAT32 formatted. To do this you will need to download and install an SD card formatter program such as SD Formatter 4 available from `https://www.sdcard.org/downloads/formatter_4/`.

Using SD Formatter 4

To use SD Formatter 4 you run the program, select your SD card from the drop-down menu, choose *overwrite format* with the *Overwrite* option as shown in figure 2.1 and then click on *Format*. It takes a while to run.



2 Select Format Option

○ Quick Format

It formats the card quickly without erasing data.

Data in the card may be retrieved after Quick Format.

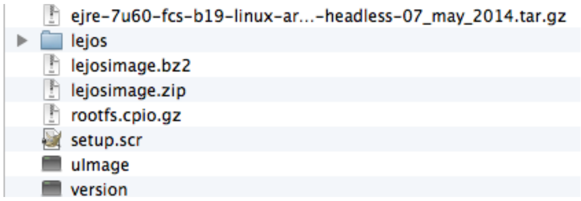
● Overwrite Format

It takes time depending on the card capacity with overwriting and erasing data in the card.

Figure 2.1: SD Formatter 4: Selecting Overwrite

2.1.3 Installing leJOS on the SD card

To install leJOS on the card unzip the zip file containing leJOS available from the Lego Rovers website (`http://legorovers.csc.liv.ac.uk`) then copy the files to the newly formatted SD Card. The card should contain the files shown in figure 2.2. Eject the SD card from your computer.



ejre-7u60-fcs-b19-linux-ar...-headless-07_may_2014.tar.gz
lejos
lejosimage.bz2
lejosimage.zip
rootfs.cpio.gz
setup.scr
uImage
version

Figure 2.2: leJOS files on the SD Card

2.2 Installing the SD card on your EV3 Brick

Put the SD card in the SD slot in your EV3 brick (shown in figure 2.3) and switch the robot on by pressing the dark grey button in the centre of the brick (shown in figure 2.4). The first time you do this it will take a while to start up as it boots from the SD card.

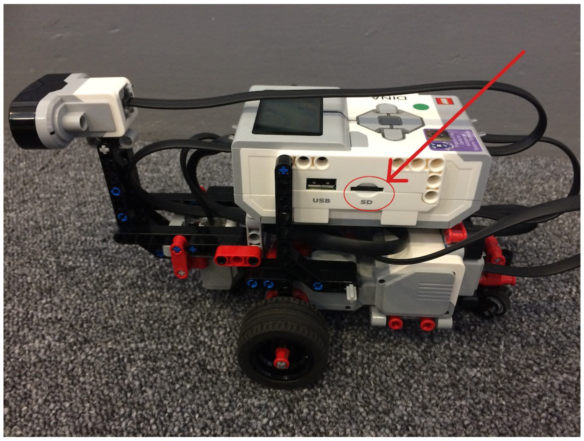


Figure 2.3: The SD slot on the EV3 brick

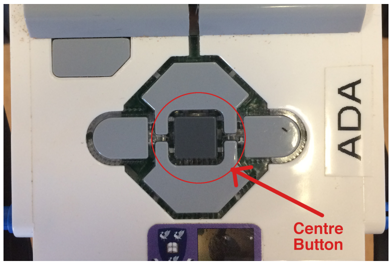


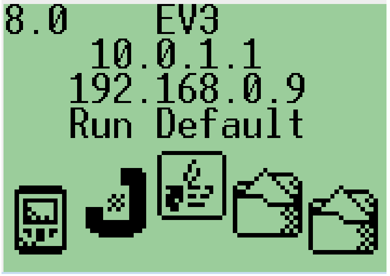
Figure 2.4: The Centre Button on the Brick

If, at any point, you want to revert to using the Mindstorms language that comes with the EV3 then you simply remove the SD card and reboot your robot.

2.3 Naming your EV3 robot (Optional)

By default your EV3 brick will be named “EV3” and will appear as such in the list of Bluetooth devices available for connection. There is no need to change this but if you will be working with several robots you may wish to give them all individual names since each of your Android devices will need to connect to a different robot.

To do this you will need to use the menus on the EV3 robot. The robot’s screen should look like figure 2.5. This is the top leJOS menu. You can navigate through the menu items by pressing the left and right buttons (shown in figure 2.6) on the front of the EV3 brick.



```
8.0     EV3
   10.0.1.1
  192.168.0.9
  Run Default
```

Figure 2.5: Default leJOS EV3 Screen

Navigate to the System menu shown in figure 2.7. Press the central button to enter the menu and then navigate through the menu items until you reach

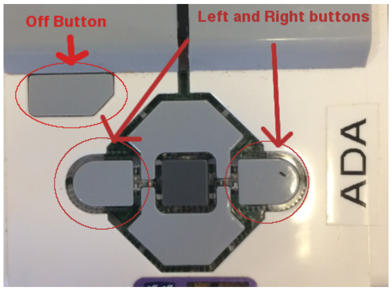
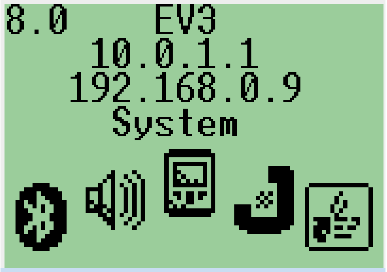


Figure 2.6: Left and Right Menu Buttons and the Off button

Change Name (figure 2.8). Press the central button again. You will then enter a screen where you can change the name by navigating around the alphabet using the buttons on the front of the EV3. This screen is shown in figure 2.9. You can switch between upper and lower case letters using the up arrow symbols on this screen, and enter spaces using the rectangle symbol. Once you have finished you press the tick symbol to enter the new name.



```
8.0     EV3
   10.0.1.1
 192.168.0.9
    System
```

Figure 2.7: leJOS System Menu

Once you have set a new name you will need to reboot the EV3 brick. Turn it off using the button on the left below the screen (shown in figure 2.6) then selecting the tick using the left and right buttons. It will take about 5 seconds to shut down. Once it has shut down, turn it on again using the centre button. The top menu should now show the EV3's new name.

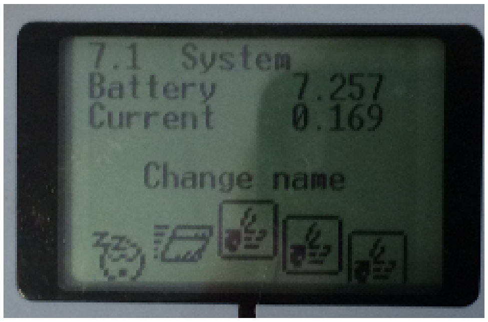


Figure 2.8: The Change Name Menu Item

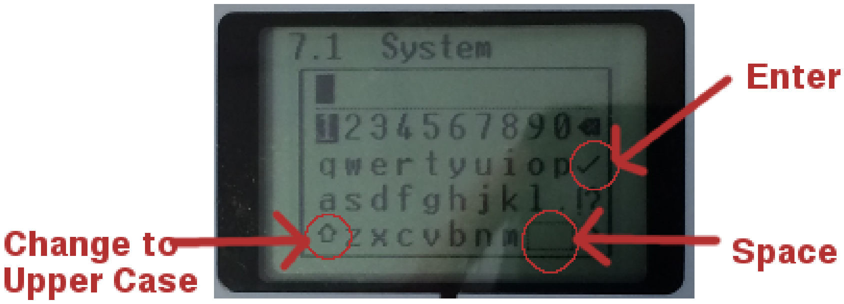


Figure 2.9: Name Input Screen

2.4 Installing and Configuring Lego Rovers on your Android tablet

2.4.1 Android Requirements

In order to install the Android application, your Android device must be Android 4.4 (KitKat) or higher and have a screen size of at least 7 inches. You can find your Android version by going to Settings and choosing *About device.*

Important: Your device must also be capable using an external paired Bluetooth device for Internet access ("Bluetooth Tethering"). Only some

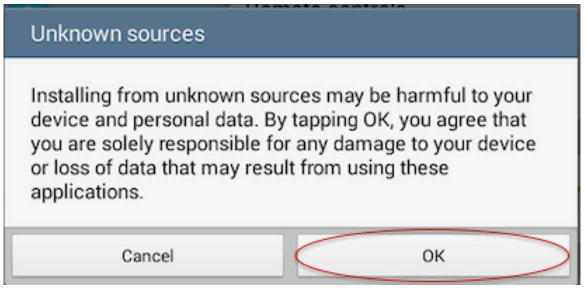
Android devices allow this (for instance the Google Nexus 7 with Android 6 does not allow Bluetooth tethering, although there are workarounds. Google Nexus 7's running Android 4 or 5 do allow tethering). This is complicated by the fact that these tablets generally will allow a mobile phone to Bluetooth tether to them and so may list Bluetooth tethering as a feature even though they only allow it in one direction. It isn't necessarily easy to determine whether a tablet will allow tethering short of pairing it with a Bluetooth device (such as a 3G or 4G enabled smartphone) and then attempting to use that device for Internet access.

Samsung Galaxy 5s running Android 6 do allow Bluetooth tethering.

2.4.2 Before Installation

Before you can install the application you first need to allow your Android Device to install applications from unknown sources. This is because, at the time of writing, the application is not available from the play store.

To do this first go into settings. Then choose security. Check unknown sources and choose OK on the information that appears (shown in figure 2.10).

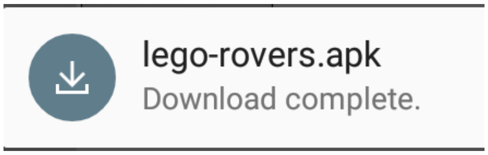


Installing from unknown sources may be harmful to your device and personal data. By tapping OK, you agree that you are solely responsible for any damage to your device or loss of data that may result from using these applications.

Figure 2.10: Installing from Unknown Sources

2.4.3 Installation

Download the Lego Rovers application from the downloads section of the Lego Rovers website. There are two apps available. One for the Lego Rovers Home Edition and one for the Lego Rovers Education Edition. Click on the relevant link and your tablet will download an apk file.



lego-rovers.apk

Download complete.

Figure 2.11: Download Message

Figures 2.11 and 2.12 show the download complete notification and installation message. You tap on the download complete notification in order to install the application.

2.4.4 Pair your Android Device and LEGO Rover

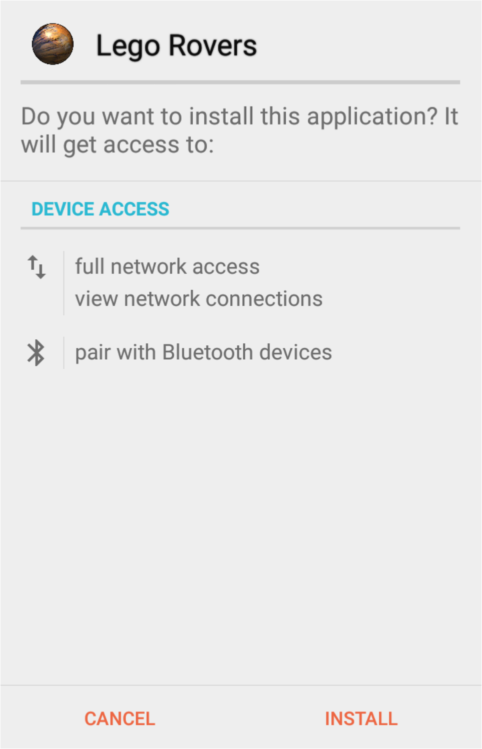
For your Android device to communicate with your EV3 robot they need to be *paired*.

For the robot to be paired with the Android device both WiFi and Mobile data must be switched off. Switch on the robot and wait for it to chime to show it is ready for use. On your android tablet go to settings and turn WiFi off and turn Bluetooth on. Once Bluetooth is on, you should be able to see your robot's name (EV3 by default). You may need to instruct the tablet to scan for Bluetooth devices. Tap the robot's name. You will probably be asked for a passkey or PIN, type 1234 and wait for the robot to pair. Click OK on the message that appears.

It should be noted that it can take several attempts to pair an android tablet and an EV3 robot, so you may need to attempt pairing several times. If the pairing repeatedly fails then it may be that your tablet does not allow Bluetooth tethering for internet access.

2.5 Building the robot

Building the physical robot is an activity that a child can be involved with. For this reason separate build instructions are available separately from the download section of the Lego Rovers website. Follow the instructions in the the appropriate *Build Instructions* manual for your set (either Home or Education).



Lego Rovers

Do you want to install this application? It will get access to:

Figure 2.12: Installation Message

2.6 Calibrating the Sensors

The Lego Rovers activity converts information from the robot's sensors into beliefs: obstacle, water or path. There are default settings for these beliefs but you may need to alter these, particularly if you are working in a small space, or if the light levels are low. In all cases you change the calibration by first connecting to the robot (as explained in the workbook) and then using the menu (located by tapping the three lines in the top left corner of the screen (shown in figure 2.15)). The menu is shown in figure 2.13). Open the Settings Pane (shown in figure 2.14).

2.6.1 Calibrating the Obstacle Belief

The obstacle belief is set to appear when the distance sensor drops below 0.4 metres (i.e., 40cm). In most spaces this works well, giving the robot plenty

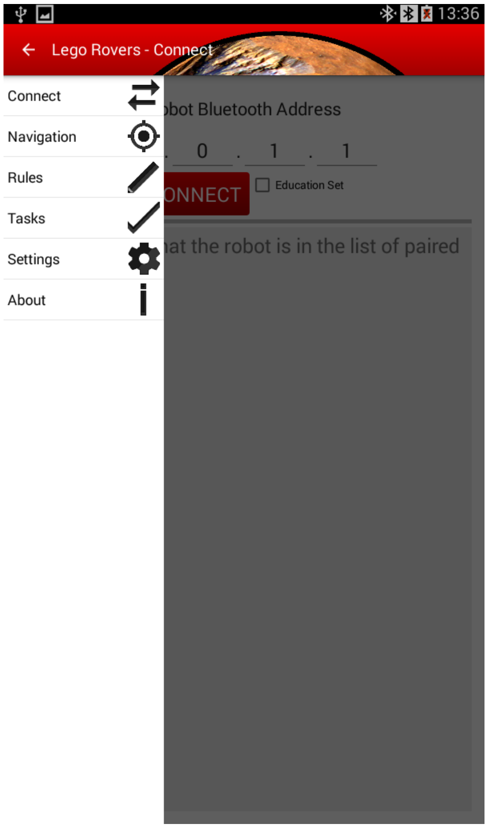
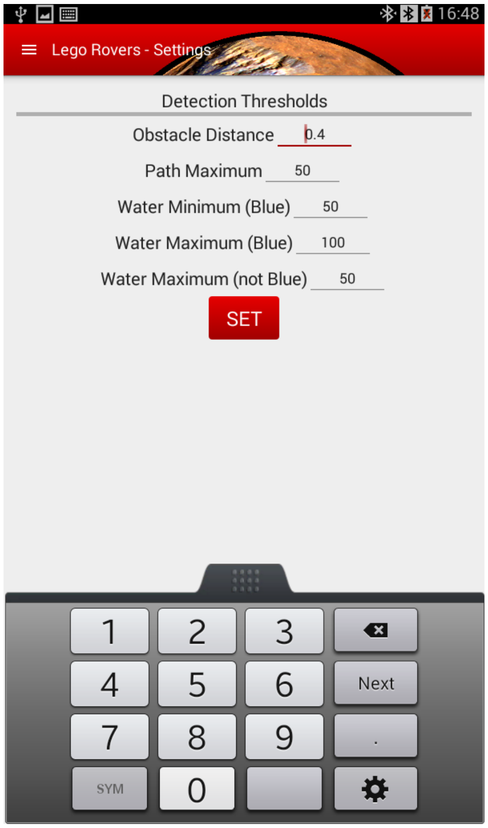


Figure 2.13: The Menu

of time to react to the appearance of obstacles, especially if being used at a fast speed. However in small or crowded spaces this can result in the robot continually detecting obstacles. In this case the obstacle distance setting can be lowered. We do not recommend lowering much below 0.2 (20cm) since this will result in more collisions.

2.6.2 Calibrating the Path Belief

The path belief is set to appear when all three RGB light values are below 50. This is actually quite generous and in many situations lowering this to 20 or 10 will be more than sufficient for detecting black lines. If you are getting a lot of false positives for paths then place the EV3 robot on the path while looking at the Navigation Pane and note the red, green and blue values of



Lego Rovers - Settings

Detection Thresholds

Obstacle Distance 0.4

Path Maximum 50

Water Minimum (Blue) 50

Water Maximum (Blue) 100

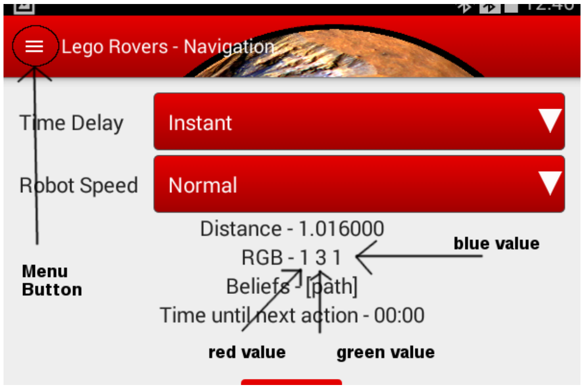
Water Maximum (not Blue) 50

Figure 2.14: Settings Pane

the sensor (this is shown in figure 2.15). Then move to the Settings Pane and change the Path Maximum setting so it is only just larger than the highest of the three colour values.

2.6.3 Calibrating the Water Belief

The water belief is particularly sensitive to light levels. When placed over a blue colour the blue reading from the RGB sensor should be higher than the other two colours (red and green). If your robot is not detecting water when placed on a blue colour (or is detecting water when placed on black) then place the EV3 robot on the path while looking at the Navigation Pane and note the red, green and blue values of the sensor, then place it on “water” and note the new red, green and blue values, lastly place it over the white



Time Delay Instant

Robot Speed Normal

Distance - 1.016000
RGB - 1 3 1
Beliefs - [path]
Time until next action - 00:00

Figure 2.15: Calibrating the Red, Green and Blue values

paper and note the values once more. Then move to the Settings Pane and change the Water Minimum (Blue) setting so it is higher than the blue value when placed on the path, but lower than the blue value when placed on water. Change the Water Maximum (Blue) setting so that it is higher than the blue value when placed on water but lower than the value when placed on the white paper. Lastly change the Water Maximum (not Blue) setting so that it is just higher than the larger of the red and green values when looking at the blue paper.

2.7 Starting the Activity

Every time you run the activity you will need to do the following:

1. Switch on the Lego Robot. It will take about a minute to start up.
2. Switch on the android tablet.
3. In the android tablet settings:
   (a) disable WiFi (and mobile data, if applicable)
   (b) enable Bluetooth
   (c) from the list of available Bluetooth devices select the one with the name of your robot and select the option that allows you to use it for internet connectivity.

4. Start the Lego Rovers app.

The workbook available from the Lego Rovers website contains a sequence of activities, challenges and projects that a child can use to guide them through the application.

2.8 Shutting Down

Once the activity is finished you can exit the app on the Android device. You turn your EV3 off using the button on the left below the screen (shown in figure 2.6). Then selecting the tick using the left and right buttons.

Chapter 3

Lesson Plans

This chapter provides two lesson plans, the first appropriate to Key Stage 2 and the second to Key Stage 3. The lesson plans assume that the class is being led through the activity by a teacher or other demonstrator and is not using the workbook. Each lesson plan is designed to last about an hour but can be easily shortened by omitting some of the later stages. The suggested timings for each stage are approximate and may need to be varied depending upon the size of the group. In general the activity works best if small groups (2-4) children work with each robot but this does mean that attention must be paid to appropriate turn-taking behaviour, especially with younger age groups.

3.1 Key Stage 2 Lesson Plan

The children should be divided up into groups of approximately 2-4, each with a robot. If resources and space allow equipment such as boxes and cardboard can be made available with which to build obstacle courses and the like.

Introduction (2-3 minutes) The demonstrator introduces themselves and the activity. They should explain that the aim is to find out about *autonomous systems* – that is computer systems which interact with the "real world" and which have to make decisions. The group will do this by investigating an example of such systems – the planetary rovers that explore the Moon and Mars.

Teleoperation (10 minutes) Each group is presented with a robot and tablet that are already connected with the Lego Rovers app running and open on the Navigation Pane. They are told to work out how to control the robot.

Time Delay (5-10 minutes) The demonstrator explains about the time delay from Earth to the Moon and Mars and that this causes trouble for driving Planetary Rovers. They then set the Time Delay on each of the tablets to the Earth to Moon setting and allow the children to experiment driving with the delay. The demonstrator concludes this stage of the activity by emphasising that it is difficult to control a robot with a time delay and so planetary rovers need to be aware of the world about them and to make decisions for themselves in order to avoid hazards.

Sensors (5 minutes) The demonstrator explains that the robot has sensors in order to detect its environment. They draw the children's attention to the distance readout on the app, explain how the distance sensor works (using echoes or reflections depending upon the sensor used), and encourage them to experiment with it, for instance by pointing the robot at the ceiling (to get a reading of "infinity" when there is no echo/reflection) and by placing their hands a few centimetres from the sensor to get a low reading.

Rules (5-10 minutes) The demonstrator explains that the robot is programmed to *believe* there is an obstacle when the distance sensor drops below 0.4. They draw the children's attention to the beliefs readout on the Navigation Pane. They then ask the children to think about what they would like the robot to do when it detects an obstacle, in terms of a sequence of the commands the children already have in the Navigation Pane. The demonstrator then travels around each group and uses the Rules Pane to set up a rule for each group to experiment with.

Line Following (5 minutes) The demonstrator shows one of the robots executing the line following algorithm using the Tasks Pane and explains how the algorithm works.

Close and Reflection (10-15 minutes) In the remaining lesson time the children may continue exploring the robots, ask questions of the demonstrator, review what they've learned, or discuss planetary rovers and autonomous systems. The workbook contains additional projects and challenges that can form starting topics for discussion. Other topics can include:

- What school subjects are useful if you want to work with robots (maths, science and (where possible) engineering).
- What kinds of robots or autonomous systems might appear in future (driverless cars, helper robots for the elderly, search and rescue robots)?
- What sorts of things does an autonomous robot need to be able to do as well as reacting to beliefs appearing and disappearing (plan ahead, adapt to things breaking/changing, learn about the world and the people it interacts with, recognise objects from cameras)?
- Do robots need to know the difference between right and wrong?

3.2 Key Stage 3 Lesson Plan

The children should be divided up into groups of approximately 2-4, each with a robot. If resources and space allow equipment such as boxes and cardboard can be made available with which to build obstacle courses and the like.

Introduction (2-3 minutes) The demonstrator introduces themselves and the activity. They should explain that the aim is to find out about *autonomous systems* – that is computer systems which interact with the "real world" and which have to make decisions. The group will do this by investigating an example of such systems – the planetary rovers that explore the Moon and Mars.

Teleoperation (10 minutes) Each group is presented with a robot and tablet that are already connected with the Lego Rovers app running and open on the Navigation Pane. They are told to work out how to control the robot.

Time Delay (5 minutes) The demonstrator explains about the time delay from Earth to the Moon and Mars and that this causes trouble for driving Planetary Rovers. They then set the Time Delay on each of the tablets to the Earth to Moon setting and allow the children to experiment driving with the delay. The demonstrator concludes this stage of the activity by emphasising that it is difficult to control a robot with a time delay and so planetary rovers need to be aware of the world about them and to make decisions for themselves in order to avoid hazards.

Sensors (5 minutes) The demonstrator explains that the robot has sensors in order to detect its environment. They draw the children's attention to the distance readout on the app, explain how the distance sensor works (using echoes or reflections depending upon the sensor used), and encourage them to experiment with it, for instance by pointing the robot at the ceiling (to get a reading of "infinity" when there is no echo/reflection) and by placing their hands a few centimetres from the sensor to get a low reading.

Rules (5-10 minutes) The demonstrator explains that the robot is programmed to *believe* there is an obstacle when the distance sensor drops below 0.4. They draw the children's attention to the beliefs readout on the Navigation Pane. They explain how to find the Rules Pane and ask each group to create a rule for the robot when it detects an obstacle.

Obstacle Avoidance (5-10 minutes) The demonstrator explains about events (the appearance or disappearance of a belief) and that the robot can have a rule for what to do when an obstacle appears and when an obstacle disappears. They challenge the children to create a program which will allow the robot to move continuously around the space avoiding obstacles. If guidance is needed to help the children then the Avoid Obstacle task can be shown in the Tasks Pane. The children should be challenged to create an avoid obstacle program with only two rules each with only one action.

Line Following (5-10 minutes) The demonstrator shows one of the robots executing the line following algorithm using the Tasks Pane and challenges the children to reproduce this behaviour with rules.

Close and Reflection (10-15 minutes) In the remaining lesson time the children may continue exploring the robots, ask questions of the demonstrator, review what they've learned, or discuss planetary rovers and autonomous systems. The workbook contains additional projects and challenges that can form starting topics for discussion. Other topics can include:

- What school subjects are useful if you want to work with robots (maths, science and (where possible) engineering).
- What kinds of robots or autonomous systems might appear in future (driverless cars, helper robots for the elderly, search and rescue robots)?
- What sorts of things does an autonomous robot need to be able to do as well as reacting to beliefs appearing and disappearing (plan ahead, adapt to things breaking/changing, learn about the world and the people it interacts with, recognise objects from cameras)?
- Do robots need to know the difference between right and wrong?

Chapter 4

Sample Answers and Explanations

Throughout the activity the user is set several programming challenges. This section discusses aspects of these in more detail, provides sample answers and discusses common pitfalls.

4.1 Connection

Lego Rovers starts up with the Connect Pane showing. If you are using the activity with a class of children then you may want to handle connecting the robot and tablet yourself and start the children in the Navigation Pane. Each time the activity is started the Android device needs to be set up to use the EV3 robot for Internet access and then the app must be connected to the robot via the Connect Pane. This is covered in chapter 2 section 2.7.

Although a keyboard pops up in the Connect Pane with the option to change the IP address (10.0.0.1) used, there should never be any need to do this and the keyboard can simply be removed by tapping the android device's back button. There is a checkbox to set depending upon whether the Education or Home EV3 set is being used. The key difference is whether the robot is built with an ultrasonic sensor (Education Set) or an infra-red sensor (Home Set).

4.2 Teleoperation

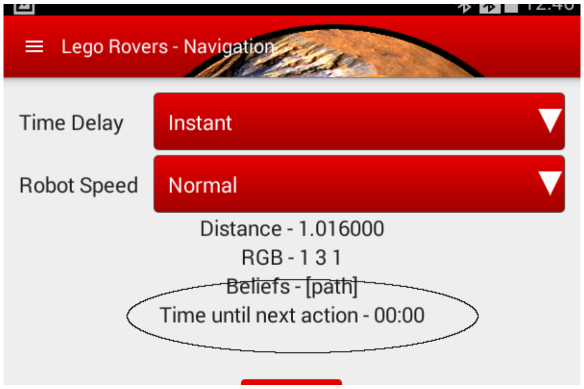
The workbook starts in the Navigation Pane where the user is encouraged to *teleoperate* the Lego Rover using 9 commands which consist of Stop (in the centre), four commands to move or turn the robot a fixed distance, and four commands to move or turn the robot until the command is interrupted (either by Stop or another command). Children can change the robot's speed and see the difference it makes to control.

This section should be fairly straightforward for most children.

4.3 Time Delay

The workbook then asks the user to investigate teleoperation of the robot at various time delays representing a satellite phone call to Australia and the time it takes a radio signal to reach the Moon. Some (especially younger) children will need some time to understand what is happening with the time delay and it may be necessary to get them to count to one or two after pressing the controls in order for them to understand that they need to wait for the robot to respond.

A time delay of three minutes to Mars is provided but in general it isn't a good idea to use this since it is longer than most children have the patience to wait. A "time until next action" number is displayed (figure 4.1) so someone can watch until their action takes place.



Lego Rovers - Navigation

Time Delay: Instant

Robot Speed: Normal

Distance - 1.016000

RGB - 1 3 1

Beliefs - [path]

Time until next action - 00:00

Figure 4.1: Time Until Next Action

The Lego Rovers app does not "queue" actions. If a child presses the buttons several times in quick succession only the last two actions will be performed. This is done to prevent long sequences of actions being built up while children understand how the time delay works, but it does have the unfortunate side effect of preventing a child who wants to "plan ahead" from being able to queue up a quick sequence of actions.

Project Answers: The section in the workbook ends by asking children to investigate the time delay from Earth to Mars. This time delay varies depending upon the relation of Earth to Mars in their orbits. It can be as little as 3 minutes, but in general it is about 20 minutes. The situation is further complicated by the fact that, because Mars has an atmosphere that can interfere with communications, commands are normally sent to a satellite orbiting Mars. This satellite then waits until it is over the rover before relaying the commands. If the satellite or rover is on the far side of Mars then it can take several hours for communication to be possible. In these situations NASA engineers generally wait for everything to be lined up before sending a message.

Roughly once every two years Mars and Earth are in *opposition* which means that the Sun is in between them and communication is difficult or impossible. Opposition lasts for roughly two weeks and no new instructions can be sent to Mars rovers during that time.

NASA has a website dedicated to Mars exploration which covers all this material.

Teleoperation and Time Delays Hopefully the user will realise that it is difficult to tele-operate a planetary rover with a time delay. In fact it *is* possible to learn to operate one with the delay caused by Earth to Moon distances, but it is impossible for a robot on Mars. For these robots engineers plan in advance what a robot should do (e.g., move forward a metre then turn right) and send those commands to the robot and wait for new information to come back after the move has been executed. This makes achieving anything, such as moving to a rock and taking a sample from it, very time consuming.

4.4 Sensors

The workbook next discusses sensors. This is a good opportunity to check that path and water detection are calibrated correctly (if not follow the instructions in 2.6).

You may be asked about the sensors on the existing Mars Rovers and/or on driverless cars. These are good projects to set for independent investigation by the children if it seems appropriate.

Mars Rovers Planetary rovers have a number of sensors. In particular they have cameras both high up on the neck (the NavCams) and low down near the wheels (the HazCams). The NavCam is used by engineers back at NASA for planning what the rover should do next while the HazCams are used by the rover itself to detect and avoid hazards.

Driverless Cars Driverless cars use sensors very like sophisticated versions of the distance sensor used on the Lego Rovers. Most driverless cars use laser based sensors with a 360° view so they can tell exactly where obstacles are around them.

4.5 Rules

The Lego Rovers can be programmed from the app using the Rules Pane. A rule consist of

1. A checkbox enabling or disabling the rule.
2. A *trigger event* that the rules reacts to. The event has two parts:
   (a) A *belief* (obstacle, path or water)
   (b) Whether the belief has appeared or disappeared
3. A list of up to three actions the robot may take once the rule is triggered.

4.5.1 Rule Triggers

Rules are triggered by an event such as an obstacle appearing or an obstacle disappearing. This is mostly simple but can cause confusion when a rule is triggered but the belief that appeared to trigger the action has not disappeared (in the case of it being triggered by something appearing) or reappeared (in the case of be being triggered by something disappearing). For instance, consider a robot approaching a corner. When the obstacle appears a rule with one action, Left a bit is triggered. The robot turns but the obstacle remains – it hasn't disappeared or appeared so no extra rule is fired. This is shown in figure 4.2.

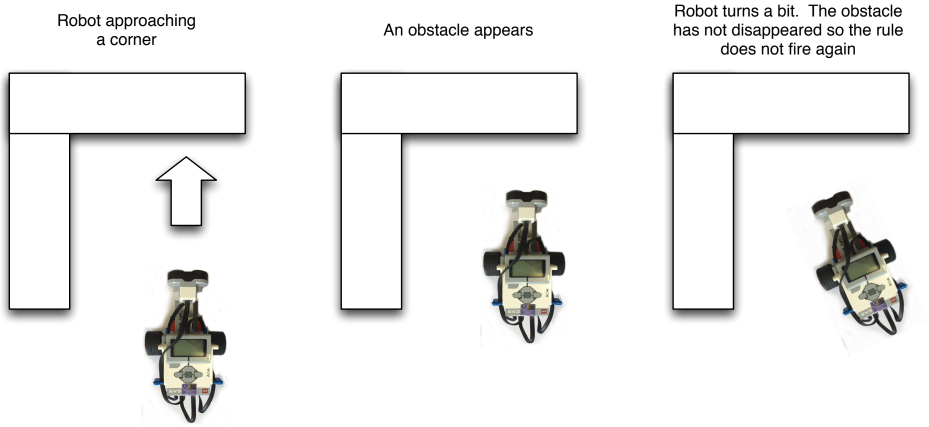

Robot approaching a corner
An obstacle appears
Robot turns a bit. The obstacle has not disappeared so the rule does not fire again

Figure 4.2: An Obstacle Rule firing when the Robot is in a Corner

It is important to remember that a rule is triggering *only when a belief appears or disappears* **not** *while a belief holds.*

Question: What happens when two rules are active? The workbook has the child experiment with two rules using the same trigger to see what happens. Rule 1 will fire but rule 2 will not. This is true for all four rules. If two or more of them use the same trigger event then only the first will be used.

4.5.2 Rule Actions

A rule specifies that the robot take a sequence of actions when it fires. It is possible to select up to three actions *but there is no problem using fewer actions*. Most of the actions are those that were available in the Navigation Pane with the addition of 90° turns left and right and two "Arc" turns that can be used when programming line following behaviour.

Interruptable Actions Some of the actions are *interrupatable* – that is the action continues until it receives a new instructions (such as Stop). These actions are Forward, Back, Left, Right, Arc Left and Arc Right. It is important that these action **are always the last** in a sequence. If they are not the last then they will be immediately interrupted by the next action in the sequence. The rule book gives the child an opportunity to experiment with this to see its effects.

4.6 Tasks

The Tasks Pane contains three pre-programmed tasks. The first two translate into programming challenges for the user of the workbook and the last challenges them to think a little about the design of programming languages and environments.

4.6.1 Avoiding Obstacles

The first task is avoiding obstacles. This pre-programmed task has the robot turn while an obstacle is being detected and then move forward when the obstacle vanishes.

Challenge: Programming Obstacle Avoidance It is possible to replicate the obstacle avoidance behaviour using just two rules each with only one action. The first rule is "If an obstacle has appeared then Left" (or Right) and the second is "If an obstacle has disappeared then Forward".

Many children attempt to program obstacle avoidance using just one rule, something to the effect of "If an obstacle has appeared then turn Right 90° then move Forward". The problem with this rule can be demonstrated by placing the robot in a corner so that even after it has turned through 90 degrees there is still an obstacle in front of it. Thinking in terms of reacting to events instead of fixed sequences of actions is quite unfamiliar to children, especially those who have already experience a little bit of programming, and so it is often worth taking a little time talking them through the ideas here.

4.6.2 Following Lines

The second task is a simple line following algorithm. For this task the robot turns in one direction when it detects a path and turns in the other direction when it doesn't detect a path. The combination of these two behaviours is a zig-zagging motion as the robot moves around one edge of a thick black line. A line following mat is available from the Lego Rovers website (http://legorovers.csc.liv.ac.uk) which can be used for this task. This mat consists of a thick black loop and the robot can move around either the outer or the inner edge. The mat is available in three sizes for printing, A2, A3 and a version to be printed on two A4 pieces of paper which can then be taped together.

Challenge: Programming Line Following It is possible to program line following in the Rules Pane in a similar way to the programming of obstacle avoidance (i.e., as two rules each with only one action). The regular turn actions, Left and Right, will not work for line following since they make the robot turn on the spot. Instead the Arc Left and Arc Right actions will need to be used. These actions make the robot turn around one while which remains stationary, causing the robot to move forwards as it turns.

The final program should consist of two rules: "If a path has appeared then Arc Left" (or Arc Right) and "If a path has disappeared then Arc Right" (or Arc Left).

4.6.3 Finding Water

The final task is an algorithm to find water. This algorithm consists of short forward movements with built in obstacle avoidance behaviour. These movements stop if either water is detected (in which case the robot stops) or a path is detected. If a path is detected then the robot will line follow (without avoiding obstacles) until it finds water. A find water mat is available from the Lego Rovers website (http://legorovers.csc.liv.ac.uk) which can be used for this task. This mat consists of a thick black loop with a blue rectangle at one end to represent the water. The mat is available in three sizes for printing, A2, A3 and a version to be printed on two A4 pieces of paper which can then be taped together.

This task is particularly sensitive to calibration of the sensors (see section 2.6) and, in particular, the robot may well accidentally detect water while it is attempting to follow the path. In some situations careful calibration can avoid this, but in some lighting situations it can not be avoided.

4.7 Final Challenges

The workbook concludes with three final projects and challenges.

Research Planetary Rovers The first project encourages the child to research planetary rovers. There are a number of excellent websites which discuss these (we provide references to some of these in chapter 5). At the time of writing there are two active planetary rovers, Mars Opportunity and Mars Curiosity both operated by NASA. The European Space Agency (ESA) plans to send its own rover, ExoMars, to Mars in 2020. The first planetary rover was Lunokhod 1 which was sent to the Moon by the Soviet Union in 1970.

Research Line Following There are many versions of this algorithm and it is possible to find videos and discussion of it in many places on the Internet. With three sensors the robot could be made to follow the line much more accurately. The sensors could be placed in a row and the robot programmed to move forward as long as the central sensor sees black and the other two see white. If one of the side sensors detects black then it should turn the robot until only the central sensor sees black once more.

Why can't Lego Rovers Rule program Find Water There are several reasons why the Find Water algorithm can't be programmed using the Lego Rovers rules. Firstly it needs rules for obstacles appearing and disappearing, paths appearing and disappearing and water appearing (5 rules) when only 4 are available. More importantly it needs to know not to turn away if it encounters an obstacle while following the path than while looking for the path and not to start turning if an obstacle appears once it has found water. The issues raised by this are discussed a little further in section 4.8.

4.8 Artificial Intelligence Programming and Beliefs

The style of programming used by Lego Rovers is a form of Artificial Intelligence programming. This kind of programming is based around *rules* that *fire* in particular circumstances rather than on devising fixed sequences of steps. A key simplification in the Lego Rovers rules is that a rule depends only upon the event that triggers it where most artificial intelligence languages allow the programmer to place conditions on rules which must also be true before they fire. For instance, when finding water there might be a condition on the rule for "if an obstacle has appeared" which adds "on condition you are not following a path".

Within artificial intelligence programming many languages use the ideas of beliefs to represent the program's view of the world. In Lego Rovers we develop these beliefs by *abstracting* the information that comes from sensors. Many artificial intelligence programming languages also use *goals* which represent things the robot is trying to achieve (such as finding water). The appearance and disappearance of goals can be used to trigger events, just like the appearance and disappearance of beliefs can and goals can appear in rule conditions as well.

Chapter 5

Resources

There are a number of challenges and suggestions for further exploration in the workbook. We list here some resources that may assist with further exploration.

5.1 Planetary Rovers and Space Science

NASA has an excellent set of resources dedicated to the Mars Curiosity Rover aimed at children. These can be found at `http://mars.nasa.gov/participate/funzone/`. From there it is possible to explore all of NASA's extensive Mars dedicated web pages which contain a wealth of images, data and explanations aimed at all levels.

The European Space Agency (ESA) are planning to put a Rover, Exo-Mars, on Mars in October 2016. They have a website devoted to this at `http://exploration.esa.int/mars/`. There is also an ESA website aimed at children at `http://www.esa.int/esaKIDSen/` with a range of information and activities covering many aspects of Space science.

The UK Space Education office, ESERO-UK, has a website dedicated to providing resources for educators on all aspects of Space Science at `https://www.stem.org.uk/esero`.

5.2 Artificial Intelligence

The style of programming based on beliefs and events that is used in the Lego Rovers activity comes from artificial intelligence. Computer Science 4 fun has a good set of resources on artificial intelligence at http://www.cs4fn.org/ai/.

5.3 Lego Mindstorms and Programming

There are numerous resources available to assist children in programming with the EV3 Mindstorms language. http://ev3lessons.com is a good website containing lessons and tutorials for all ability levels. You can easily use the Mindstorms language with your EV3 by removing the Lego Rovers SD card from it.

LEGO also run an annual competition aimed at schools and home education groups which can provide a focus for design and programming activities with the EV3. This is the First Lego League and information about it can be found at http://www.firstlegoleague.org.

As a more general introduction to programming for children then the Scratch system, https://scratch.mit.edu, is highly recommended.

There are also several national and international organisations aimed at encouraging children to program which maintain online resources and run clubs and events. These include CoderDojo (https://coderdojo.com) and Code Club (https://www.codeclub.org.uk).

5.3.1 Advanced Programmers (leJOS and Java)

The Lego Rovers app uses a Java based system called leJOS. This is available for free from http://www.lejos.org where some documentation for it can also be found. The source code for the Lego Rovers application can be found at https://github.com/legorovers. In general we would not recommend either leJOS or the Lego Rovers source code to beginners at programming, but they may be of interest to parents or teachers with a strong programming background and an interest in developing an application of their own.

Chapter 6

Troubleshooting Guide

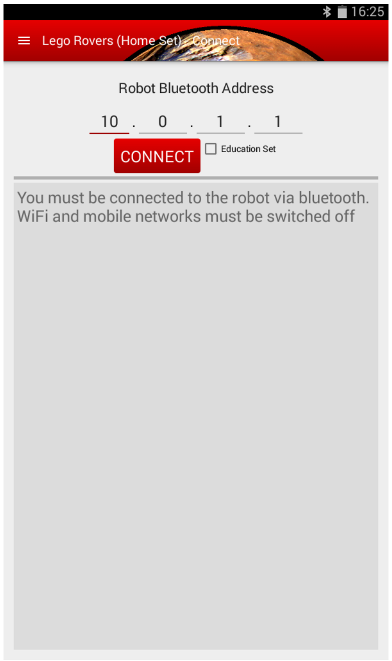
This chapter gives an overview of common issues encountered when running Lego Rovers and how to overcome these.

6.1 EV3 and Android Tablet won't Pair

It can often take several attempts to get an EV3 robot to pair with an Android device so persevere. If pairing persistently fails then it is possible that it is because your Android device does not allow Bluetooth tethering, or that there is an issue with the EV3's built-in Bluetooth. If you have a different Android device available, even if it is not suitable for running the Lego Rovers app, attempt pairing the EV3 with that device which may help determine if the problem is with the EV3 robot or the Android device.

6.2 Connect Pane: You must be connected to the robot via Bluetooth. WiFi and mobile networks must be switched off

If your android device is not using the paired EV3 for internet access then the error message shown in figure 6.1 will be displayed when you attempt to connect Lego Rovers to the device. Return to the Settings app on your Android device and check that WiFi and mobile are switched off and that you are using the EV3 for Internet access.



Robot Bluetooth Address

10 . 0 . 1 . 1

You must be connected to the robot via bluetooth.
WiFi and mobile networks must be switched off

Figure 6.1: Connection Error Message

6.3 Connect Pane: java.io.invalidClassException

Occasionally the Lego Rovers app will fail to connect properly to the EV3 robot and this error will be displayed. Often this can be fixed simply by attempting to connect a second time. However if the error repeats several times it may be that the sensors in the EV3 have either crashed or failed to disconnect from Bluetooth. In this case follow the instructions for recovering from a Bluetooth crash in section 6.4.3.

6.4 Lego Rovers Unresponsive or Crashed

There are a variety of situations that may cause the Lego Rovers app to crash or stop responding. The most common of these are the loss of the Bluetooth connection between the Android Device and the Lego Rover, and over rapid operation of menus, buttons and text fields (particularly in the Settings Pane).

6.4.1 Lego Rovers Quits Unexpectedly

In many situations Lego Rovers will shut down cleanly and can simply be re-opened and the connection to the EV3 can be restarted. However sometimes the app will fail to disconnect correctly from the EV3. If the lights on the EV3 sensors remain on, even though the application has quit, then this has happened. If this is the case follow the instructions in section 6.4.3.

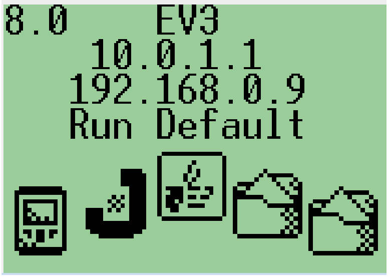
6.4.2 Lego Rovers/EV3 Unresponsive to commands

If the EV3 stops responding to commands from the Navigation Pane then it is probable that the Bluetooth connection between the Android device and the EV3 has been lost. To check this, see if the values coming in from the distance and colour sensors shown in the Navigation and Rules Panes have also have stopped changing. If you attempt to change to a different pane then the app will crash. If this is the case follow the instructions in section 6.4.3.

If data is still coming in from sensors, then it is possible that the EV3's motors are not connected properly. See section 6.8 for a discussion of this.

6.4.3 Recovering from a Bluetooth Crash

To recover from a loss of Bluetooth connection between the EV3 robot and your Android Device, you will need to reset the robot. To do this you will need to use the menus on the EV3 robot. The robot's screen should look something like figure 6.2. This is the top leJOS menu. You can navigate through the menu items by pressing the left and right buttons (shown in figure 6.3) on the front of the EV3 brick.



```
8.0    EV3
  10.0.1.1
 192.168.0.9
 Run Default
```

Figure 6.2: Default leJOS EV3 Screen

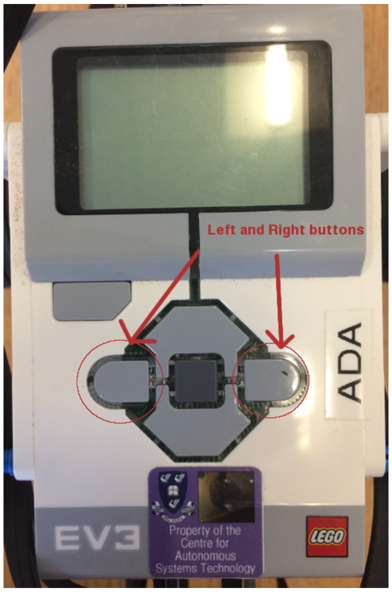
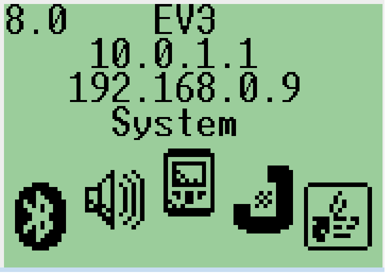


Figure 6.3: Left and Right Menu Buttons

Navigate to the System menu shown in figure 6.4. Press the central button to enter the menu and then navigate through the menu items until you reach **Reset** (figure 6.5). Press the centre button to reset. The lights in the sensors should switch off and any motors should cease – sometimes a Java error will appear but this can be ignored. It should now be possible to restart the Lego Rovers application on your Android device and reconnect to the EV3.



```
8.0     EV3
  10.0.1.1
 192.168.0.9
   System
```

Figure 6.4: leJOS System Menu

Occasionally a leJOS reset will fail and it still will not be possible to reconnect the Android device and EV3. In this case it will be necessary to

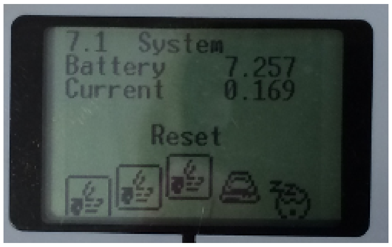


Figure 6.5: The Reset Menu Item

switch the EV3 off and on again.

Very occasionally the leJOS menu itself will be unresponsive and pressing the EV3 buttons will do nothing. In this case it is unfortunately necessary to dismantle the robot sufficiently to allow the battery or batteries to be removed. This will shut down the system which can then be rebooted by replacing the battery or batteries and pressing the centre button.

6.5 Robot moving without being told to

If you are in the Navigation Pane or the Tasks Pane and the robot is behaving unexpectedly – for instance stopping, reversing or turning when not instructed to do so using the buttons in the Navigation Pane, or moving off the path it is following in Tasks then it may be that some rules have been left enabled in the Rules Pane. Use the menu to move to the Rules Pane and check that all four rules have been disabled by unchecking the On/Off button.

6.6 Rules not Firing

There three reasons why rules may not fire.

1. Firstly the On/Off checkbox may not have been ticked and so the rule is not enabled.

2. Secondly the sensors may not be calibrated correctly. See section 2.6 in order to correct this.

3. Lastly it may be that the rule has already been triggered and has not yet been deactivated. See section 4.5.1 for an explanation of this.

6.7 Robot can't detect Paths/Water

The sensors are not calibrated correctly. See section 2.6 in order to correct this.

6.8 Robot is Turning incorrectly/one or more Wheels don't work

If the EV3 doesn't move at all, or moves incorrectly (e.g., only one wheel turns) when being operated from the Navigation Pane. Then it is possible that there is something wrong with the wheels on the robot.

1. It is possible that a wheel is obstructed in some way and can not physically turn. You can check this by moving the wheel by hand on the robot and seeing how much resistance there is. This is generally caused by some mistake when building the robot and can be fixed by disassembling the wheel and referring to the build instructions for the robot.

2. It is possible that the wires connecting the motors with the EV3 brick may be loose or damaged. Check they are all connected firmly at both ends.

3. Lastly the motors may be connected incorrectly. The left motor should be connect to port C and the right motor should be connected to port B.

6.9 Information from sensors is wrong

The sensors supplied by LEGO with the EV3 are quite noisy so you should expect them to be somewhat error prone and that the values will change even if the robot isn't moving. However if the readings from sensors shown in the Navigation, Rules and Tasks Panes are very wrong or blank then:

1. It is possible that the wires connecting the sensors with the EV3 brick may be loose or damaged. Check they are all connected firmly at both ends.

2. It is possible that the senors are connected incorrectly. The distance sensor should be connected to port 2 and the colour sensor should be connected to port 3.

3. If just the distance sensor is effected then it is possible that you are using the Home version of the Lego Rovers app with an ultrasonic sensor or the Education version with the infra red sensor. Use the menu to bring up the Connect Pane and look to see if the Education Set checkbox is set correctly (it should be checked if you are using the ultrasonic sensor and unchecked if you are using the infra-red sensor). If it is incorrect, disconnect the robot, set the checkbox correctly and then reconnect.

6.10 90° degree turns are not 90° degrees

The 90° turns are based on calculations involving the diameter of the robot wheels and the distance they are apart. This means they are only approximate angles rather than exact angles. It may be possible to make them more accurate by adjusting the distance between the EV3's wheels, but it is probably best just to live with the fact that they are only "roughly" a quarter turn.

Index