Spencer User Guide

Contents

1 Introduction **2**

2 Main Features **2**

3 Structural Information **2**

3.1 Specification of the robot 2
3.2 Limitations of the robot 2

4 Usage Information **2**

4.1 General usage 2
4.2 Turning Spencer on 3
4.3 Swapping batteries 3

5 Software Overview **3**

5.1 Setting up the server 3
5.2 Installing the App 4
5.3 Talking to Spencer 4
5.4 Stairway to heaven 5
5.5 Upwards and onwards 5

6 Troubleshooting **6**

6.1 Server startup issues 6
6.1.1 `FileNotFoundError`, `Remote I/O Error` 6
6.1.2 `PhidgetException` 6
6.1.3 `Sensor data is still invalid after 2 seconds` 6

1 Introduction

This document is a user guide for a stair climbing robot called Spencer built by team 4 during the System Design Project course. This guide includes the structural information, usage information, software overview, and a troubleshooting guide.

2 Main Features

Spencer's main features are:

- Able to carry 500g of load up and down stairs.
- Self-aligns to stairs while climbing.
- Controlled by a user friendly Android App.
- Climbs up a step in under 20 seconds.
- Climbs down a step in under 15 seconds.

3 Structural Information

3.1 Specification of the robot

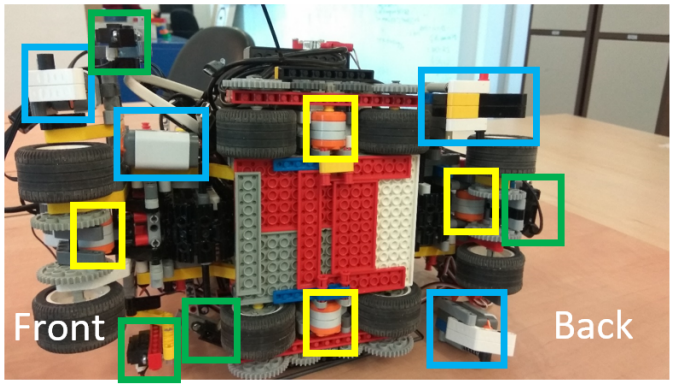

Figure 1: Structural information of the robot

As shown in Figure 1, the robot is made up of:

1. Four infrared sensors (highlighted in green).
2. Four touch sensors (highlighted in blue).
3. Six motors:
   (a) Four motors (highlighted in yellow) are responsible for moving forward and backward.
   (b) The two motors in the middle section are also responsible for turning left and right.
   (c) Two motors at the top of the robot (not shown in the figure) drive the front and back lifting mechanism.
4. Raspberry Pi, motor board, Phidget board, 4-port USB hub, rotary encoder board, two rotary encoders, power bank, battery pack.

3.2 Limitations of the robot

1. The stair must be at least 33cm deep so that the robot can fit on it comfortably.
2. The maximum stair height the robot can climb up and down is 17 cm.
3. Maximum carrying capacity of the robot is 1kg, but 500g is recommended.
4. From the default position, the front section of the robot can only move upwards and the back section can only move downwards. This means that the robot can only climb up stairs while facing the stairs and climb down stairs with the back facing the stairs.
5. Spencer cannot climb stairs that have left or right turns, nosing and no riser.

4 Usage Information

4.1 General usage

- Do not spill liquid on the robot and any container with liquid should be properly sealed.
- The battery life is approximately 1.2 hours.
- The climbing speed of the robot is noticeably slower when the batteries are low on charge.
- When going downstairs, ensure the back of the robot is perpendicular to the stairs.

- When going upstairs, ensure the robot is facing less than 45°away from the stairs. Otherwise, the robot's self-aligning software becomes unreliable.

4.2 Turning Spencer on

Before turning on Spencer, please ensure that the batteries are fully charged. If not, you will need to swap them (see section 4.3 on Swapping batteries). The micro computer in the Raspberry Pi should start automatically after you insert the micro USB cable into the Raspberry Pi and connect the white cable to the power bank. To start using Spencer, see Section 5.

4.3 Swapping batteries

To swap out the batteries:

1. Gently lower the robot onto its side and remove the four red pegs on the underside of the robot. After the pegs are removed, remove the bottom plate (see Figure 2).
2. To remove the batteries, detach the red and black wire that is connected to the top of the AA battery pack.
3. To remove the power bank, disconnect the black and white wires connected in the centre of the robot and then pull the power bank out.
4. Swap the batteries in the battery pack and recharge the power bank.
5. Reattach the red and black wire to the battery pack. Then reinsert the power bank.

> **Note:**
> The power bank should be inserted on the side closest to the front of the robot so that the top of it is beside the motor board. It may require some adjustments to get it back into position.

6. Reattach the bottom plate and place the 4 red pegs back to secure the bottom.
7. Place Spencer upright again and insert the white wire into the 1A port of the power bank and the black wire into the 2.1 A port. Spencer will turn on automatically if the steps have been followed correctly.

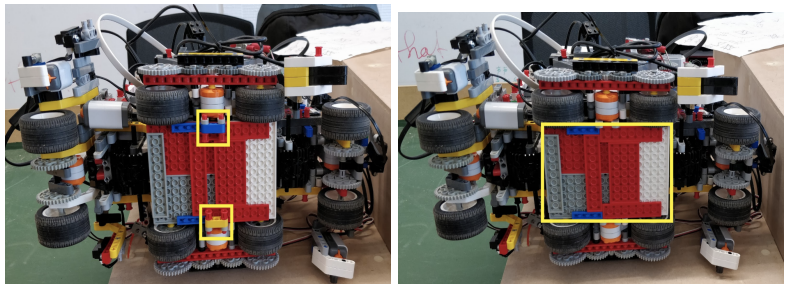

Figure 2: Locking Pegs and Baseplate

5 Software Overview

Spencer's codebase is split into two sub-projects: the Android app which receives commands from user and sends commands to Spencer, and the server code, which runs on the Raspberry Pi and receives and executes commands. The Raspberry Pi is also responsible for controlling and using sensors to navigate the steps.

5.1 Setting up the server

In order to set up the server to run on the Raspberry Pi, one must acquire the sources and install all additional dependencies. First, ensure Spencer is connected to the internet. Once set up, the source repository must be cloned and dependencies installed, below are the commands to run on terminal to install server-side code:

```
# Download the source code
$ git clone https://github.com/jackhorse1000/SDP4.git && cd SDP4
# Install pipenv and dependencies
$ pip3 install pipenv
$ pipenv install
```

Once installed, Spencer's server should be started:

```
$ cd src
$ pipenv run ./server.py
```

This should display debugging messages and then inform you the IP address and port the server is currently running on.

If you do not see such a message, or the program crashes, please refer to the troubleshooting section. Otherwise, continue and set up the app.

5.2 Installing the App

To install the app on your phone you will need an Android device. The app is still in development and so it is not available on the app store.

```
# Download the source code
$ git clone https://github.com/jackhorse1000/SDP4.git && cd SDP4/App
$ ./gradlew build
```

The resulting APK can now be found at `app/build/outputs/apk/debug/app-debug.apk`. Please connect your phone via USB and copy this file onto the device. The APK is supported on devices with Android 4.4W (API level 20) and above.

Note:

You may need to enable developer mode on your Android device, in order to allow installing APKs from untrusted sources.

5.3 Talking to Spencer

When you first open Spencer's app, you will be greeted with screen similar to that in Figure 3.

At the top of the screen, you can see Spencer's current connection state. In order to get Spencer up and running, you'll need to connect to him. Press "Connect to the robot" to go ahead and do that - make sure that you're connected to the WiFi first.

If everything is set up correctly, you should be shown a screen like Figure 4. If this is not the case, please consult the troubleshooting section.

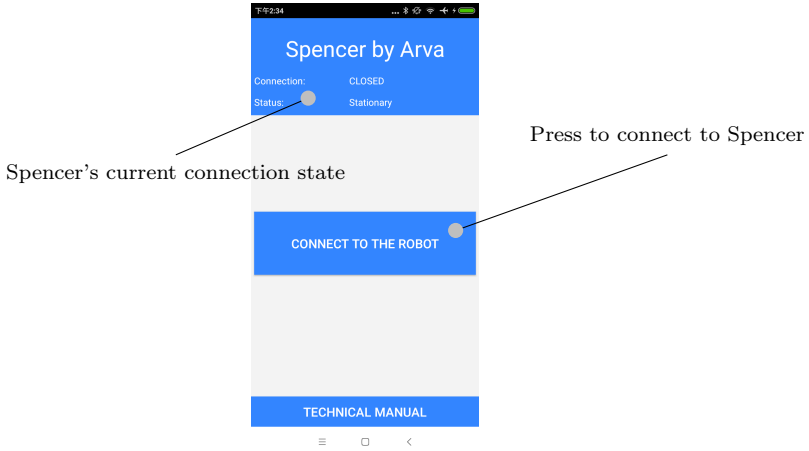

Figure 3: The startup screen for Spencer

5.4 Stairway to heaven

Now that you're connected to Spencer, you should be able to go ahead and start controlling him. As you can see in Figure 4, the app's screen is split into several sets of buttons:

Stop all movements Bring Spencer to an immediate stop. This action may terminally interrupt the stair climbing progress, and so should only be used during an emergency.

Recover initial state Reset Spencer to his initial state as seen in Figure 6. Where each of the lifting mechanisms are at their default position.

Forward, Backward, Turn left and Turn right Drive Spencer forward or backward as well as turn left or right on the spot.

Climb upstairs and Climb downstairs Instruct Spencer to start going up or down stairs.

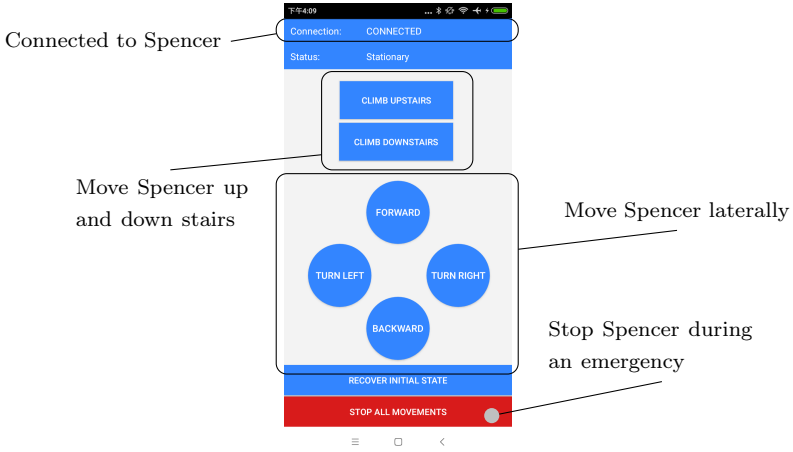

Figure 4: Controlling Spencer

Let's get started with something simple: place Spencer on a flat surface, and use the middle 4 buttons to drive him around. If you experience any issues with Spencer's movement (i.e. jerkiness, periodic slowness), please consult the troubleshooting section.

In order to climb a stair, move Spencer until his front is facing the staircase. On the app press the "Climb upstairs". The self-alignment software allows for a 45°margin of error in his angle of approach. Spencer will now continue climbing the staircase until he has reached the top or until the "Stop all movements" button is pressed.

5.5 Upwards and onwards

While Spencer is mid ascent or descent the user should refrain from pressing any other buttons except the 'Stop all movements' command in case of emergencies. After he has reached the top or bottom of the staircase he will automatically stop and wait for further instructions. Now you have the tools to control Spencer. If you experience any issues or problems please consult the troubleshooting guide.

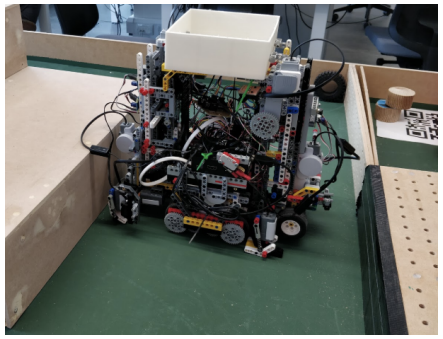

Figure 5: Spencer at the bottom of stairs, ready to climb.

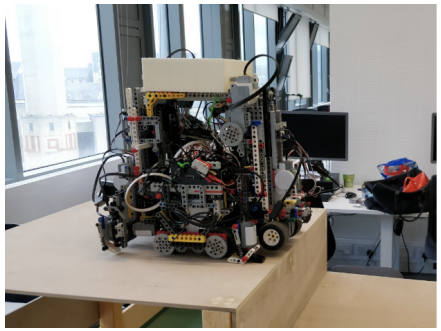

Figure 6: Spencer at the top of stairs, ready to descend.

6 Troubleshooting

6.1 Server startup issues

When the server starts up, there are several errors it may throw. The most common of these are detailed below. If you receive another error, please get in touch via help@team-arva.uk.

6.1.1 `FileNotFoundError`, `Remote I/O Error`

```
FileNotFoundError: [Errno 2] No such file or directory: '/dev/i2c-1'
OSError: [Errno 121] Remote I/O error
```

If the server crashes with this error, it implies that the motor board cannot be found, or an error occurred when sending signals to it. We recommend running `ic2detect 1` in order to determine what is currently detected to the Raspberry Pi. You should see a matrix of numbers like so:

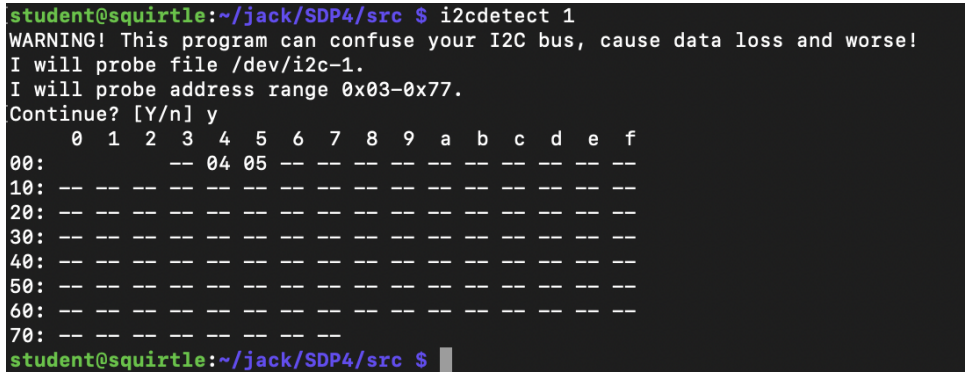

Figure 7: Running `i2cdetect`

It is key that channels 4 and 5 are shown: if they are not displayed, check your cabling to the motor board.

6.1.2 `PhidgetException`

```
Phidget22.PhidgetException.PhidgetException: 3
```

This means that the Phidget board, or some connected sensor, cannot be found. Check the wiring between the Raspberry Pi and Phidget board, and the ensure all appropriate sensors are connected to the board.

6.1.3 `Sensor data is still invalid after 2 seconds`

```
OSError: Sensor data is still invalid after 2 seconds.
```

Some unknown error occurred when starting the attaching to sensors, meaning that no sensor data was received. Spencer will refuse to start in these circumstances. It should be sufficient to run the server again, though if it occurs repeatidly, contact help@team-arva.ukfor further information.

Problem	Potential cause	Solution
Robot cannot be connected to the app.	1. Raspberry Pi is turned off. 2. The server has stopped running. 3. The app is on a different network from the Raspberry Pi.	1. Make sure the Raspberry Pi is turned on. You should observe one or more lights blinking on the edge of the case. 2. Make sure Spencer's (Raspberry Pi's) server is running. 3. Make sure the app is on the same network as the Raspberry Pi.
Robot does not respond to the application.	1. The application has crashed.	1. Restart the application.
Robot starts moving towards the wall instead of the stairs.	1. The infrared sensors responsible for robot's self-alignment pick up the closest vertical object and self-aligns to it.	1. Stop the robot and use the app to drive and align the robot closer towards the stairs.
Robot falls over at any point during a stair climb.	1. The stairs might be unsuitable for the robot (stair surface, stair height, stair depth). 2. There might be an object on the stair in the robot's way. 3. The batteries and/or power bank might be low on charge.	1. Place the robot upright on the ground and reset the robot. 2. Make sure that there are no objects on the stairs blocking the robot. 3. Recharge the batteries and power bank.
Robot cannot lift the object it is carrying.	1. The object might be too heavy for the robot to carry. 2. The batteries and/or power bank might be low on charge.	1. Reduce the amount of weight the robot is carrying. 2. Recharge the batteries and power bank.
Front section of the robot has been lifted to its maximum height and the front section is still lower than the height of the step.	1. The step is too high for the robot to climb.	1. The robot is not usable on these stairs.
The front or back section does not lift even though the motors controlling the robot's lifting mechanisms are functional.	1. The tracks of the lifting mechanism are broken.	1. Press "Stop all movements" button and contact customer service at help@team-arva.uk.
Any of the sensors detach from the robot when the robot is in operation.	1. The sensors got caught on something when the robot was moving.	1. Please contact customer service at help@team-arva.uk.
Any other issues.	1. Misuse of the robot. 2. Extremely rare use case. 3. A cable snapped or became loose.	1. Please contact customer service at help@team-arva.uk.