Mission 13: RoboWarriors

Start date: 05 Oct 2017
Due: 13 October 2017, 23:59

Preliminaries

This mission will be done in groups of 3 or 4 students. Only one person needs to submit the group's programs on the Source Academy, with their group name and members written in a comment at the top of the submission. Other members should submit their group name and members also, but need not submit their programs. Everyone in the group will get the same grade.

As this project requires a fair amount of time and effort, you are encouraged to check out your robot kit and start working on it as soon as possible!

There are a few steps required of you.

1. Kit Checkout
2. Environment Set Up
3. Writing and testing your solutions
4. Assessment
5. Submission of Mission 13
6. Sumobot Contest
7. Kit Return

1. Kit Checkout

Only one robot kit will be loaned to each group.

Collect your kit from Mr. Chow Chin Ming at Technical Services (Ground floor, COM1).

Collection times: **4 October to 6 October** (Wednesday to Friday, week 7) Wednesday 12:00pm - 5:00pm Thursday 9am to 12pm, 1pm to 5pm and Friday 9am to 12pm, 1pm to 3pm. You **MUST** check out your kit by 6 October 2017 (Friday).

2. Environment Set Up

Hardware Setup

Instructions for a default robot design are included in the manual, which can be found in the robot kit. Try that out if you don't know where to start. You are, however, encouraged to come up with your own design!

For the mission, it should be a robot that your group has built. Sharing of the same robot for grading is not allowed.

Installing the ev3dev Image

Download the ev3dev image here: [http://www.ev3dev.org/downloads/) and use an image burner of your choice to install the image onto the microSD card issued. You will require a microSD card reader for this. The instructions for each operating system is as follows:

Windows

- Download the Win32DiskImager software from http://sourceacademy.comp.nus.edu.sg/uploads/Win32DiskImager.zip.
- Unzip it and now you have a new folder called "win32diskimager-v0.7-binary"
- If your computer has a slot for SD cards (SD to micro SD adapter needed), insert the card. If not, insert the card into an SD card reader, then connect the reader to your computer. Note: must be formatted in FAT32!
- Run the file named **Win32DiskImager.exe** (in Windows Vista, 7 and 8 right-click this file and choose "Run as administrator").
- If the micro SD card (Device) you are using is not found automatically then click on the drop down box on the right and select the micro SD card letter you just plugged in (e.g. [H:]). Note: must be formatted in FAT32!
- Be careful to select the correct drive; if you get the wrong one you can destroy your data on the computer's hard disk!
- In the Image File box, choose the .img file that you downloaded and click "Write". Note: if a warning message appears click YES.
- Your microSD card is ready to be used.

Instructions adapted from udoo

Mac

For Mac users, we recommend Etcher. Otherwise, refer to the Linux instructions to format your microSD card using the command line.

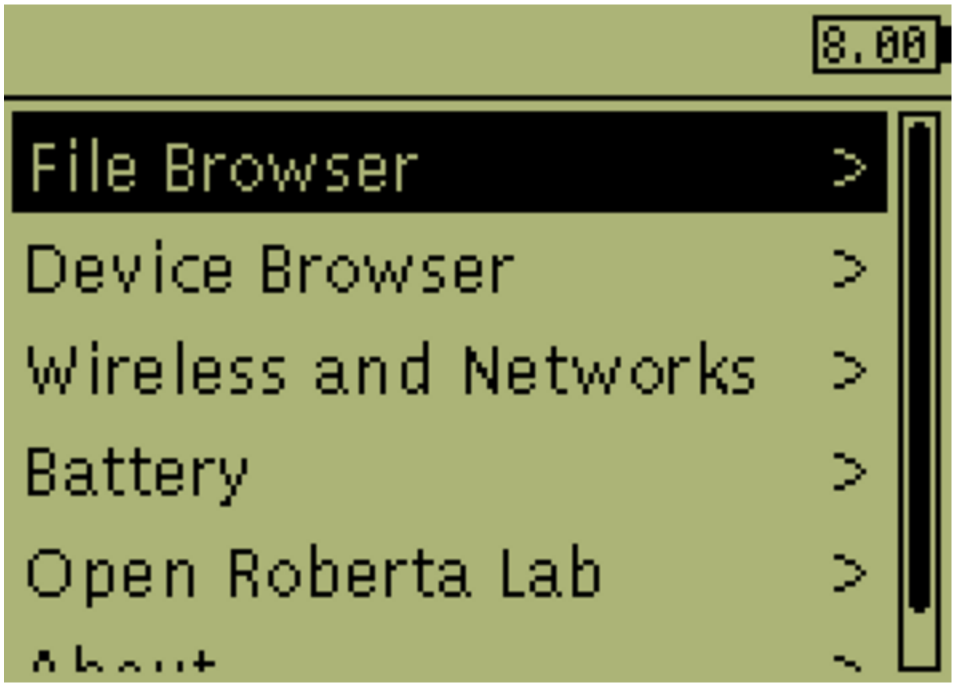
Linux

Follow the link for command line instructions to write the image above into the microSD card issued.

Software Setup

The environment that we have installed in the microSD card is a linux distribution from a project called ev3dev. You can find out more about it from the official website. For this mission, you will be writing JavaScript. Remember, the Source language is a subset of the JavaScript language. Your solutions will be written in the NodeJS environment. Your MicroSD card has already been formatted with the distro that includes `ev3dev-lang-js`, the library that you will be using in the mission.

This is what you should see when the EV3 has booted up with the MicroSD card inserted.



3. Writing and Testing your solutions

Introduction

To program on the EV3, we will be using the SSH protocol to remotely access the linux distribution on the EV3. Before you can SSH into your EV3, it has to be connected in some way for it to be remotely accessible.

How to connect

The EV3 can be connected via a Bluetooth, Wi-Fi or wired connection. However, we have had most success on the **wired connection** and we highly recommend connecting via the USB wire provided in your kit.

To connect to your EV3 via the USB wire provided, you can follow the instructions from http://bit.ly/2dkpDXh. The key is to run

```
ssh robot@ev3dev.local
```

successfully so that you are connected to your EV3. Enter your password when prompted. The default password is `maker`.

You can also use other SSH clients like PuTTY (http://www.putty.org/).

Troubleshooting

For Mac user, if you are running El Capitan or Sierra, you might not see 'CDC Composite Gadget' in the interfaces list from network configuration. You can try to connect via bluetooth instead. Tutorial can be found here http://bit.ly/2dCXYVF.

How to write programs

The first thing you need to do after successfully connect to EV3 is copying the library files into the system. Download and unzip the library files from http://sourceacademy.comp.nus.edu.sg/uploads/mission_13_node_modules.zip and run

```
scp -r path/to/your/unzipped/folder robot@ev3dev.local:./
```

You can also use other scp clients like WinSCP (https://winscp.net/eng/download.php)

Remember to put these three lines at the top of your program

```
#!/usr/bin/env node
var ev3 = require('./node_modules/ev3source/ev3.js');
var source = require('./node_modules/ev3source/source.js');
```

To run a script that you have written, execute

```
node path/to/file.js
```

To make it executable from your EV3 file browser, `chmod` the file to make it executable from the command line

```
chmod +x path/to/file.js
```

You should now be able to execute it directly from EV3.

Examples

example1.ts

```
#!/usr/bin/env node
var ev3 = require('./node_modules/ev3source/ev3.js');
var source = require('./node_modules/ev3source/source.js');

var motorA = ev3.motorA();
var motorB = ev3.motorB();

if (ev3.connected(motorA)) {
    source.alert("CONNECTED");
}

if (ev3.connected(motorB)) {
    source.alert("CONNECTED B");
}

ev3.runForDistance(motorA, 3000, 100);
ev3.runForDistance(motorB, -2000, 100);
```

example2.ts

```
#!/usr/bin/env node
var ev3 = require('./node_modules/ev3source/ev3.js');
var source = require('./node_modules/ev3source/source.js');

var color = ev3.colorSensor();
if (ev3.reflectedLightIntensity(color) > 20) {
    // Do something
} else {
    // Do something else
}
```

Documentation

The list and miscellaneous functions available in Source Academy (Week 7) are available for use. However, you will need to put the keyword `source` in front of the functions that you are normally used to.

```
// To create a pair(3, 4)
source.pair(3, 4).

// To obtain the head of a pair
source.head(source.pair(9, 10)).
```

Here is the list of all the functions available, you may use it as reference when you are working on this mission:

Motors:

- `ev3.motorA()`: returns the motor connected to the port A.
- `ev3.motorB()`: returns the motor connected to the port B.
- `ev3.motorC()`: returns the motor connected to the port C.
- `ev3.motorD()`: returns the motor connected to the port D.
- `ev3.runForDistance(motor, distance, speed)`: causes the motor (which you get from the previous functions) to rotate at the specified speed for a specific number of rotations; if you pass a negative distance, the motor will rotate backward.

- `ev3.runForTime(motor, time, speed)`: causes the motor (which you get from the previous functions) to rotate at the specified speed for a specific duration.
- `ev3.stop(motor)`: cause the motor (which you get from the previous functions) to stop rotating.

Color sensor:

- `ev3.colorSensor()`: returns the connected color sensor.
- `ev3.colorSensorRed(colorSensor)`: returns the red value read from the `colorSensor`, it's a number within the range of 0 to 1020.
- `ev3.colorSensorGreen(colorSensor)`: returns the green value read from the `colorSensor`, it's a number within the range of 0 to 1020.
- `ev3.colorSensorBlue(colorSensor)`: returns the blue value read from the `colorSensor`, it's a number within the range of 0 to 1020.
- `ev3.reflectedLightIntensity(colorSensor)`: return the percentage of the reflected light intensity.

Touch Sensor

- `ev3.touchSensor1()`: returns the touch sensor connected to the port 1.
- `ev3.touchSensor2()`: returns the touch sensor connected to the port 2.
- `ev3.touchSensorPressed(touchSensor)`: returns a value based on the amount of pressure detected by the touch sensor.

Ultrasonic Sensor

- `ev3.ultrasonicSensor()`: returns the connected ultrasonic sensor.
- `ev3.ultrasonicSensorDistance(ultrasonicSensor)`: returns the distance between the sensor and an object in centimeters (cm).

Gyro Sensor

- `ev3.gyroSensor()`: returns the connected gyro sensor.
- `ev3.gyroSensorRate(gyroSensor)`: Returns the value of the gyro sensor.

Miscellaneous

- `ev3.pause(time)`: Pauses the program for a specified amount of time in milliseconds.
- `ev3.runUntil(terminatingCondition, task)`: repeatedly executes `task()` until `terminatingCondition()` is satisfied, note that both of the two arguments are **functions**.

Part of the fun is learning how to troubleshoot. If you have difficulties, start by googling your problems. `ev3.alert()` is a useful function to output to the screen.

4. Assessment

Once you have built your robot, you will need to test your robot to make sure that it works. This mission consists of **four** tasks, which will prepare your team for the climax of this mission: the Sumobot contest!

Submission

Submit your programs for Tasks 1 to 4 by 13 October, 2359, on the Academy. Only one member of the team needs to upload the program, as stated before. Please also upload the program you have so far for Task 5; it won't be graded, but it is so your avengers can make sure that you are on the right track. It's okay if you haven't written much of it yet, and you can of course make changes before the competition starts.

As part of the submission process, you will be required to demonstrate your robot in front of an Avenger. A temporary grade will be given upon submission, which will be updated after the completion of your demo. The track for task 3 and the board for task 4 will be available for demo purposes outside SR1 over week 7 and 8. Details will be announced on the Academy, so watch out for them!

Have fun!

Task 1:

Write a program that makes your robot move forward for a distance of 10cm. In order to do that, you need to find out the speed of the robot when both motors are moving forward.

A possible way to approach this is by picking a certain duration (say 3 seconds), and then writing a program that moves the robot forward for a time equal to the chosen duration, measuring the distance you actually move. You can then use that information to compute how much time it will take to move the distance you want.

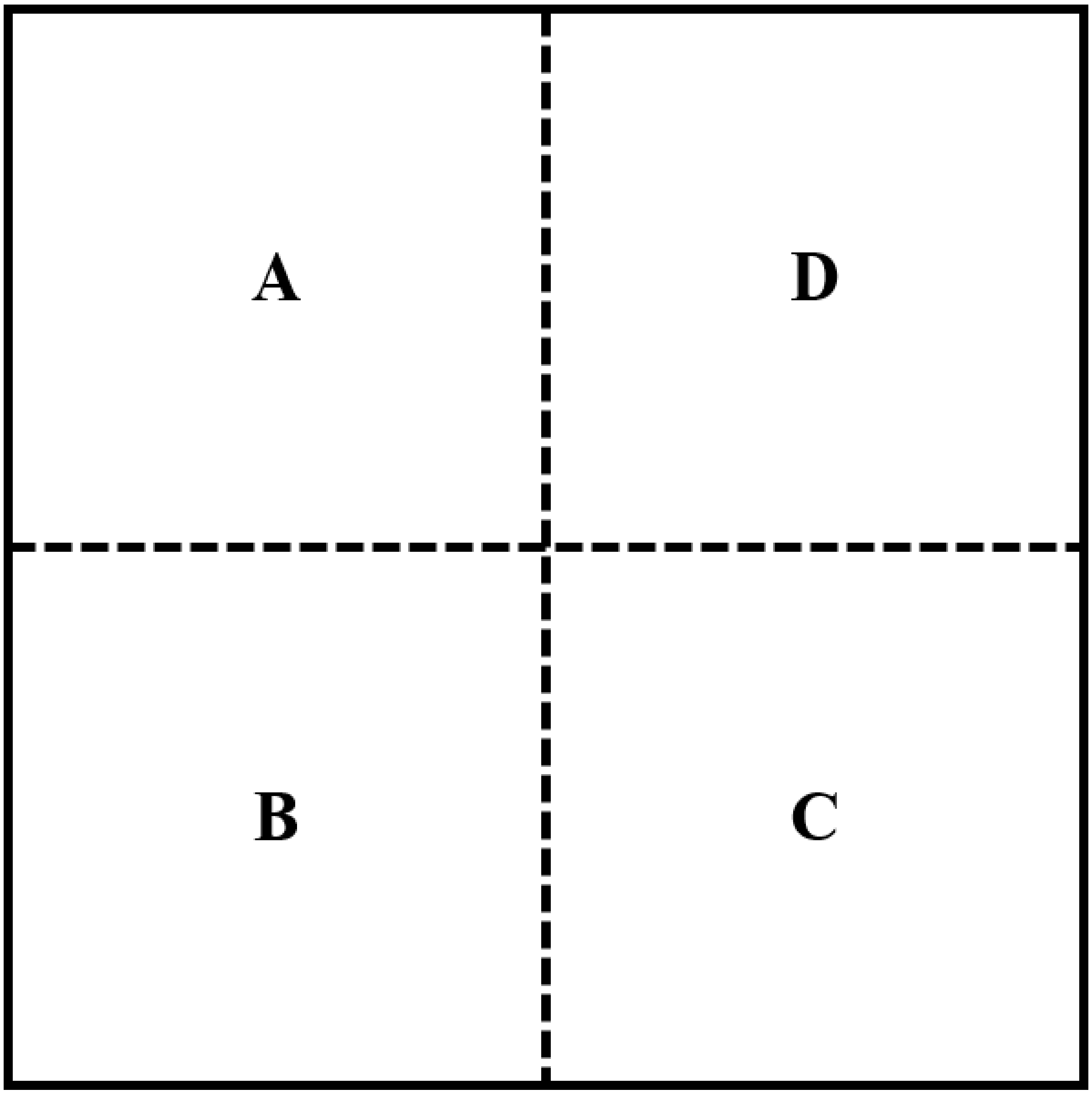
Upload your program into the robot and check that the distance is indeed 10cm.

Task 2:

Write a program that makes the robot turn 90° counterclockwise. You may choose to solve it the same way you did the previous question or implement your own solution.

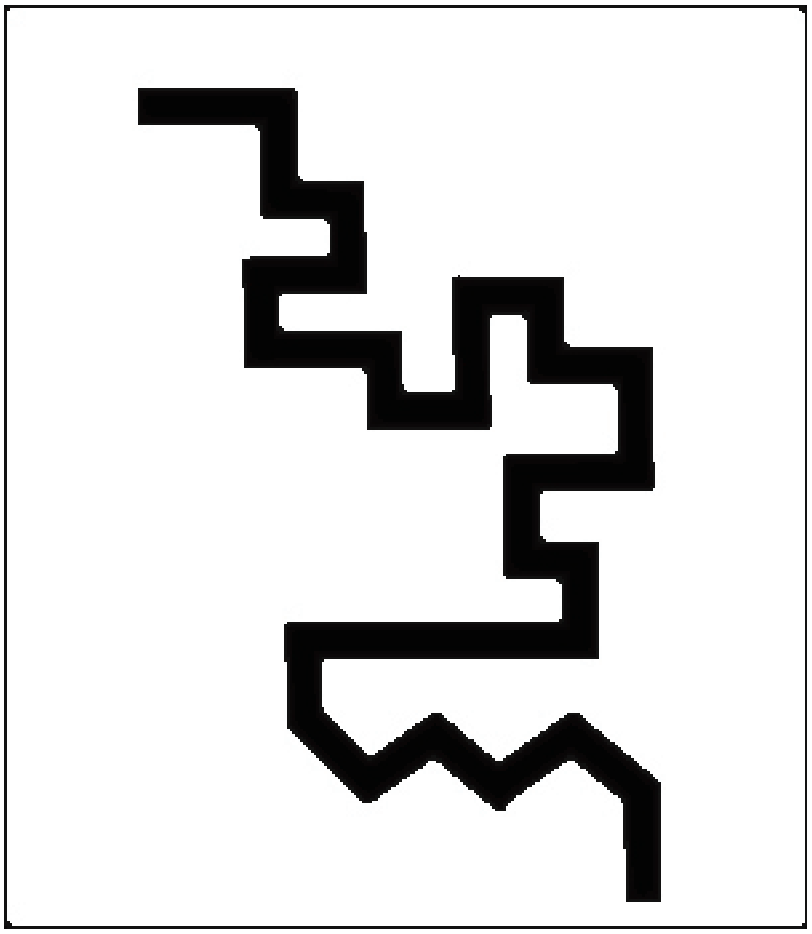
Task 3:

In this task, your robot will start from a random position on the playing field shown in Figure 1. Write a program which makes your robot make a circuit around the field without falling off the board. Your robot will be required to visit the 4 quadrants marked in Figure 1 (the quadrants will be marked with white tape on the actual board; the border will be marked with black), starting at an arbitrary position.



Task 4:

Write a program which makes your robot track a black line on an A0-sized white board. The black line will be 4 cm thick. We will have two testing tracks for this task, one of which is given in Figure 2. The other will only be made known on-site.



5. Submission of Mission 13

To submit your work to the Academy, copy the contents into the task box then saves on the mission page, then click “Submit”. Note that submission is final and that any mistakes in submission requires extra effort from a tutor or the lecturer himself to fix.

6. Additional Info for Robot Contest

You can always attempt the first four tasks with a basic robot first. The design of the robot does not need to be the one that you will be using in the contest. Upgrade the robot and add more armour to it after the grading if you wish.

You can have as many programs as you want loaded in your robot, and you are free to choose which of your programs to run in each round of the match(es), for example to deal with different kinds of opponents.

Remember to charge your battery before the robot contest.

As mentioned before, only parts from ONE LEGO EV3 set can be used when building the robot.

Besides the Sumobot contest, we will be having a pageant on the day itself. This pageant will be held just before the final round of the Sumobot contest. Teams will be given 20 seconds to show the rest of the class how beautiful/talented/creative their robots can be. You are allowed to use any additional materials for this. You may use this time to do anything: you may want to give a short speeh about your robot (e.g what your robot represents, why is it dressed up like this etc.). If your robot can sing or dance, all the better!

After the contest, pictures of the decorated robots will be placed on IVLE and you will be able to vote for your favourite robot.

Lastly, use your creativity and design an awesome robot! Make full use of the sensors and motors! Think out of the box and make history as the next Sumobot champion!

7. Kit Return

The robot should be returned any time in the following period: **20 October to 25 October** (Week 9 Friday and Monday to Wednesday, week 10) from 10am to 12pm and 1pm to 5pm. The return must be made by the same person who borrowed it. Please make sure all parts in the inventory list are present in the box, and that you haven't lost anything! You will be asked to do an inventory check before the return.