Programming Club
EV3 Robots

Based on IVR Robotics Assignment

Volker Seeker

25^{th} October 2017

1 Introduction

The aim of this problem is to build an control a robot such that it interacts with its environment. You will start off easy by going through a few tutorial exercises and then slowly build up towards using multiple sensors to navigate your robot. We will use EV3 Lego Mindstorms robots

1.1 Responsibility for the Equipment

When handing out the equipment, students are required to sign to show they have received the EV3 kit. A £5 deposit will also be taken for a key to one of the lockers. It is the responsibility of the students to take care of the equipment and to return a complete kit. The contents of the kit will be checked on return.

2 Typical Usage of the EV3 robot

The EV3 has been formatted with a version of Linux called EV3DEV (installed on the SD memory card). It can read from the sensors and can command the motors. We will use a python programming interface to control the robot. The following instructions describe

1. how to connect to the robot (aka the brick)
2. copy your program to the robot and run it
3. some tutorial exercises to become familiar with the robot.

You can find useful details about ev3dev on its project webpage: `http://www.ev3dev.org/`

and the github repository of the python language bindings: `https://github.com/ev3dev/ev3dev-lang-python`

2.1 Starting Up the robot

1. Plug USB into computer and brick
2. Press square button on ev3. It boots through to the ev3dev penguin screen and the brickman screen to the main menu (showing file browser, device browser etc)
   On DICE computer:
   (a) When the ev3 brick boots, the DICE network manager will create the ev3dev network interface with a static ip address
   (b) ssh onto the computer: ssh robot@ev3dev. Password: maker
3. You are now connected to the robot
4. (When you are finished shut down by pressing the upper right button a few times)

As a final note, do not remove the SD card from the robot as this will reset some of the settings and will require you to reconfigure them.

2.2 Typical Development Cycle:

You will develop your code (in python) in a text editor of your choice on your computer. When you want to test it on the robot, you will copy your entire code directory to the robot. This way you can keep your latest files on your computer at all times. (**Tip: use (git) version control as your code WILL break as it gets more complex**)

1. Write code on your computer. (Make sure your scripts are executable (chmod +x test.py))
2. Copy the code onto the ev3 to test: **scp -r ivr_directory robot@ev3dev:/home/robot/**
3. When you have copied your programs to the ev3 brick you can use its UI screen to run the program by selecting it through file browser. Unplug the USB cable and press the square button on the EV3 to run the program.
   You can also execute it directly from a terminal over usb to avoid having to reconnect or to easily see terminal printing ($ python test.py)
4. Its ALIVE! The robot will operate autonomously
5. To interrupt and end a program press the upper right button for about 3 seconds

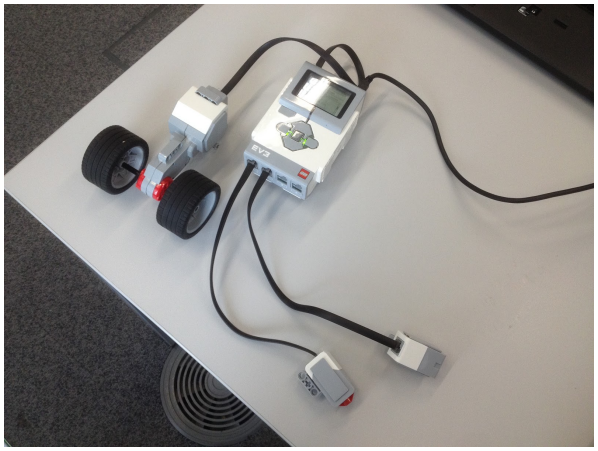
3 Tutorial Exercises

First download the code samples from this location (or use git checkout):
`https://github.com/vsee/ivr_robotics`

This contains the tutorial scripts. Put it somewhere on your local file system and uncompress it if you downloaded the archive file. You can keep working within this directory or copy the file you want to reuse for your own implementations. We recommend building reusable classes and functions. Do not implement everything in a single sequence of predetermined functions to make bug hunting easier when it eventually breaks.

3.1 Explore Basic Operation of Sensors and Motors

1. Build the simple wheeled vehicle illustrated below. Make sure to connect the switches to numbers and motors to letters, as shown (brick, wheels and two switches)
2. Connect to the robot as described in Typical Usage
3. Progressively work through the tutorials in main.py. You can run main.py from either a terminal or through the EV3s navigation UI.
   (a) Open Loop driving
   (b) Turn an LED on and off with a switch
   (c) Drive backwards and forwards using one switch to command each direction
   (d) Build your first python class to keep state of robot
   (e) Attach the Ultrasound sensor, read a set of measurements from it and write them to a text file (depending on where you attach it, you might need to adapt the sample code)

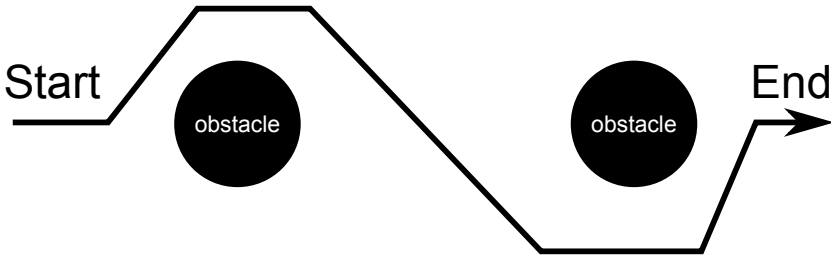


4 Problems

The following problems will introduce you to various sensors and motors. They are presented in an increasing level of difficulty, but feel free to pick and choose as you like.

4.1 Driving

1. build the robot following steps 1 - 40 of the EV3 manual
2. write a program to make it drive through an obstacle course similar to the one depicted below
3. measure the time and try to beat your highscore without bumping into any obstacles



Snippet of code to operate the motor and read its encoder value (which is the count of the number of counts it turns):

```
m = ev3.LargeMotor('outA')
m.connected
m.run_timed(time_sp=3000, speed_sp=720)
while True:
    print(m.position)
```

4.2 Colour Recognition

1. attach the colour sensor to your robot (Manual 73 - 76)
2. write a program which lets you recognise colours held in front of the sensor (you can print a message or make the robot say the colour aloud)

Snippet of code to interpret colour sensor readings. Values are between 0 and 7 with the following interpretations:

Colour	none	black	blue	green	yellow	red	white	brown
Value	0	1	2	3	4	5	6	7

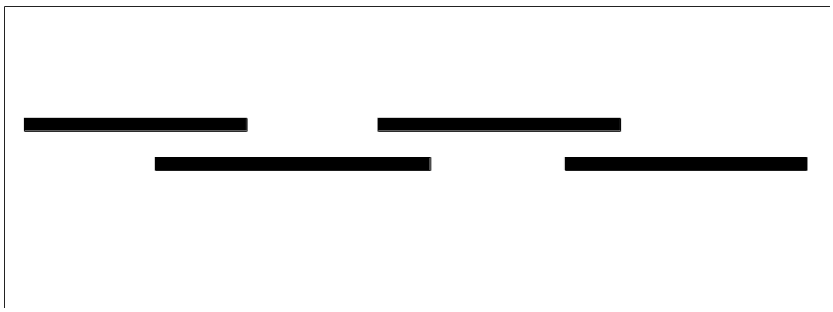
```
cs = ev3.ColorSensor(ev3.INPUT_4)
cs.connected
cs.mode = 'COL-COLOR'
while True:
    print(cs.value())
```

4.3 Driving between Lines

1. modify the colour sensor to point downwards (Manual 69 - 72)
2. make the robot drive until it finds a black line, then make it stop
3. add more complicated drive pattern, e.g. stop at the third line, increase robot speed between second and third line, ...

4.4 Driving along Lines

1. add the gyroscope to your robot (Manual 48 - 53)
2. write a program that allows your robot to drive along an arbitrary line using the colour sensor to find it and the gyroscope to turn
3. modify your program, so the robot can drive along broken lines as depicted below



Snippet of code to print the gyro reading: It returns a number in degrees that the sensor has turned:

```
g = ev3.GyroSensor(ev3.INPUT_4)
g.connected
g.mode = 'GYRO-ANG'
while True:
    print(g.value())
```

4.5 Avoiding Obstacles

1. add the ultrasonic sensor to your robot (Manual 42 - 47)
2. make your robot drive until its sensor finds a close target, then make it stop
3. modify your program to make the robot drive around and obstacle using the ultrasonic sensor and the gyroscope
4. use the medium motor to build a movable ultrasonic sensor like the one in the given video and modify your program to use the new setup
   `https://www.youtube.com/watch?v=u9SYv3LLMek`

Snippet of code to print the ultrasonic sensor reading. It returns millimetres:

```
us = ev3.UltrasonicSensor(ev3.INPUT_4)
us.connected
us.mode = 'US-DIST-CM'
while True:
    print(us.value())
```