EV3 Quick Guide

SDP Team

January 2018

1 Introduction

The aim of this guide is to allow you to quickly setup your EV3 brick and how to execute programs. The brick has a custom OS installed called `ev3dev`, with more details found here `http://www.ev3dev.org/`.

The EV3 has a variety of sensors that are connected via one of the four ports on the bottom of the brick, listed as 1, 2, 3, and 4. The top ports, listed as A, B, C, and D are used for motors. This tutorial will go through setting up motors and a few of the sensors, specifically the touch, light and ultrasonic sensors.

2 Connecting to the brick

To turn on the brick make sure that the EV3 brick is not connected to any motors or sensors, as this will significantly increase the time for the brick to boot. This guide will focus on using the brick via the `DICE` machines, and will only be concerned with connecting using a USB cable. For details on using non-`DICE` machines please see here [1], and here for Bluetooth connection [2].

The instructions for `DICE` are as follows:

1. Turn on the EV3 brick
2. Plug the USB cable into the PC port on the top of brick, and the cable into the computer
3. ssh into the brick through the terminal: `ssh robot@ev3dev`
4. The password is maker

If you are having difficulty connecting, the brick may not have its IP address setup correctly. To fix this:

1. Navigate on the brick to Wireless and Networks
2. Then go to All Network Connections
3. Click on Wired

4. Scroll to IPv4
5. Press Change
6. Change the IP address to 192.168.17.129 (if you ping ev3dev this will also say what the IP address should be)

3 First Commands

The EV3 brick can run python programs, and so can also run commands via the python terminal. Type `python3` to open the terminal. If you enter the code below

```
import ev3dev.ev3 as ev3
m=ev3.LaregMotor('outA')
if not (m.connected):
    print(''Plug a motor into port A'')
else:
    m.run_timed(speed_sp=300, time_sp=1000)
```

the motor will run for one second if connected properly. What the code is doing is firstly importing the EV3 modules, then assigning a motor, and lastly we perform a check to make sure the motor is connected.

This allows small snippets of code to be tested on the brick quickly in real time (as will be seen is useful in the next section). It is important to note that the commands the EV3 is currently running may continue even if you exit the python terminal, i.e. likely a motor is continuing to operate. If this happens open up the terminal again, assign the correct motor and use the code

```
import ev3dev.ev3 as ev3
m=ev3.LaregMotor('outA')
m.stop()
```

If you are running a program through the UI on the brick you can stop the program by pressing the top left button for 3 seconds.

4 Running Programs

The general development cycle for software development with python on the EV3 is

1. Write your code in a text editor
2. Send your files to the EV3: `scp file1 file2 etc robot@ev3dev:/home/robot`
3. Test your code on the brick by using the UI on the brick, or connect to the brick and through the terminal type: `./program.py`

4. If you ran the program via the UI and the program is in a loop, you can stop the program by pressing the top left button for 3 seconds

To be able to run the python code via the UI on the brick it needs to be made as an executable file. The bricks should be set up already to do this, but at the start of any python file that you wish to run as an executable needs to have this line of code

```
#! /usr/bin/env python3
```

There is a text editor on the brick, nano, that you can use to edit files already on the brick. This will obviously not change the code you have on your computer, meaning that any uploads of your code from the computer will overwrite the files on the brick. It's advised that any changes on the code should be done on the files on your computer and then re-uploaded to the brick to esnure consistent work copies.

`ev3dev` can also support programs written in `c++`. These programs have to be compiled first though, and requires to be done through the `ev3dev` OS. So while programs can be compiled on the brick itself this is very slow and it is advised instead to use docker to compile the programs first on your computer. Afterwards transfer the compiled programs across. Full details can be found here [3].

References

[1] `http://www.ev3dev.org/docs/tutorials/connecting-to-the-internet-via-usb/`.

[2] `http://www.ev3dev.org/docs/tutorials/connecting-to-the-internet-via-bluetooth/`.

[3] `http://www.ev3dev.org/docs/tutorials/using-docker-to-cross-compile/`.