Project Venus

# User Guide

SDP Group 15-H

February 1, 2016

## 1 Installation

### 1.1 Cloning the repository and setting up

To clone the repository, execute in terminal:

```
$ git clone https://github.com/julijonas/venus.git
```

Then create a Python virtual environment and install the libraries:

```
$ cd venus
$ virtualenv env
$ source env/bin/activate
$ pip install -r requirements.txt
```

To open the Arduino IDE, execute:

```
$ arduino
```

You'll need to add three libraries which can be found under `arduino/` in the project directory: *ArduinoSerialCommand*, *SDPArduino*, *SimpleTimer*. In order to add a library, go to:

```
Sketch -> Import Library... -> Add Library... and choose the library folder
    you want to import.
```

## 2 Overview of the robot

At the current stage, our robot, Venus, can move and kick. It moves because it has its wheels connected to the NXT motors and we can send commands that power some of those motors on and off depending which action we want the robot to perform. The kicker is also connected to another NXT motor, which allows us to control the power of the kick.

## 2.1 Use

In order to turn the robot on, connect the battery pack to the power board. In order to swap the batteries, take them out, put them to charge and insert the new batteries into the battery pack. Easy!

# 3 Running instructions

(To be changed after the first milestone)

Run the command in the terminal from the project root directory:

```
source env/bin/activate
```

Then, in order to send commands the robot:

```
python control/control.py
```

After that you should be able to send commands to the robot (make sure the RF stick is plugged in).

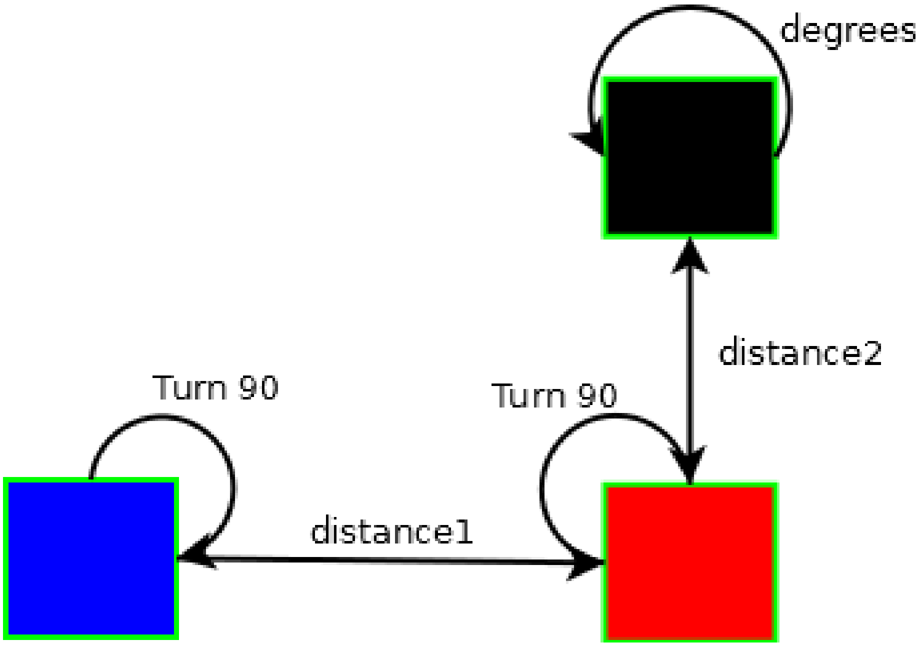
**NB** In case you make changes to Arduino code, you should upload your changes to the board!

The list of possible commands is below:

- For moving to the target in the milestone 1, there is a command 'move'.

  At the beggining the robot turns 90 degrees to the right (if "distance1" parameter is positive the robot will move to the forwards, if negative backwards). After covering the first distance it turns 90 degrees so that it faces the same way as at the start. The "degrees" parameter is amount of degrees the robot turns (positive - clockwise, negative - anticlockwise)

  ```
  move <distance1> <distance2> <degrees>
  ```



- To transfer a file through RF link to I2C port as specified in the milestone:

```
transfer <filename> <frequencyInHz>
```

- To kick the ball a distance specified in the milestone:

```
kick [50|100|150]
```

- To move forward and turn (negative values will change the direction):

```
f <distance>
c <degrees>
```

- To engage the grabber:

```
g
```

- To release the grabber and kick simultaneously:

```
x <kickStrength>
```

- To stop all the motors:

```
s
```

# 4 Troubleshooting guide

**Problem 1** When trying to upload the code to the Arduino board, you get the error message below.

```
processing.app.SerialNotFoundException: Serial port '<port>' not found. Did
   you select the right one from the Tools > Serial Port menu?
```

**Solution** It can't detect the Arduino board. Make sure it's connected to the computer through the RF stick or the cable. If it is, disconnect and connect again, or power the board off and on again. The Arduino IDE is only able to program the Arduino when the serial interface of the RF stick or Arduino itself is registered as `/dev/ttyACM0` on the PC, that is, zeroth device. Also, make sure you are not running 'python control/control.py' or have not opened the serial interface any other way when you're uploading the changes.

**Problem 2** When sending commands the robot doesn't do anything and you're sure it should move.

**Solution** Power the Arduino board off and on again.