Robots Integration

Juan Pedro López
Oriol Orra
Ismael Rodriguez
Carles Oró

Documentation Arduino – Motor

The goal of this work is to be able to control the velocity of a motor. In order to achieve this objective it is used the equation of a simple line, which relates the velocity (rpm) of the motor to its position. The next step will be to connect the motor to a webcam and move it in different velocities depending on the position of the target element. For instance, if it is wanted to follow a human face, the closer it is the faster the webcam has to move.

As it is mentioned before, to determine the desired parameters it is used the equation below.

$$y = m * x + b$$

It is known that the motor can reach a maximum speed of 60rpm, and this is obtained at 0º. In the other hand, at 90º there is no velocity.

The slope of a line can be determined by:

$$m = \frac{90-0}{0-60} = -\frac{9}{6}$$

Afterwards, it is isolated the parameter b as follows:

$90 = m * 0 + b$ → b = 90

To sum up, the equation of a line should look like this.

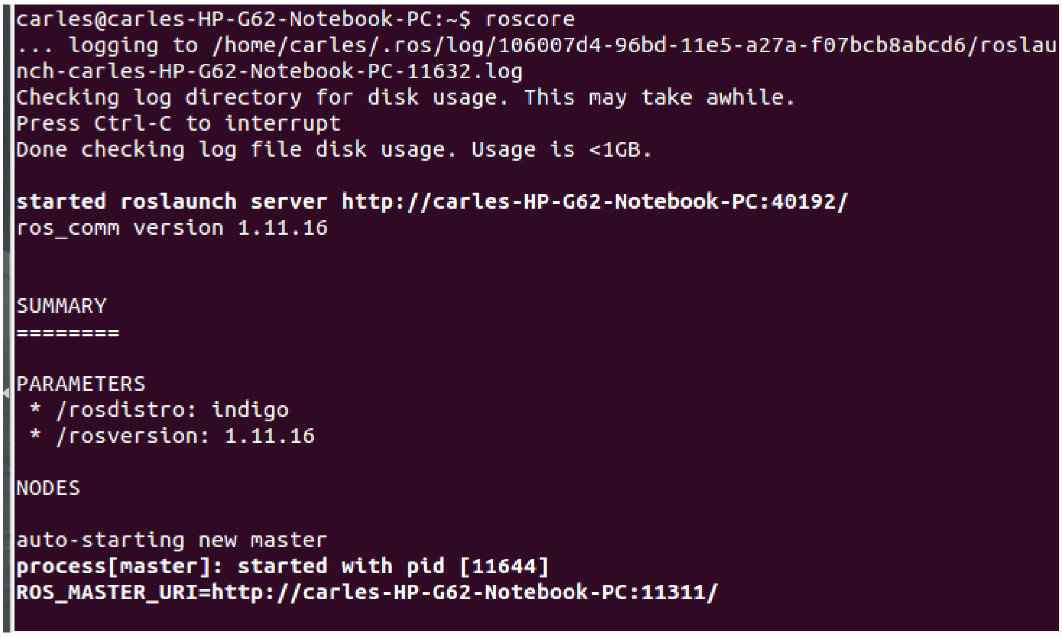
$$y = -\frac{9}{6} * x + 90$$

Where:
y = the input angle of the motor in degrees
x = velocity of the motor (rpm)

The used program works with ROS, then just with entering the desired value of the velocity it will automatically translate to degrees.

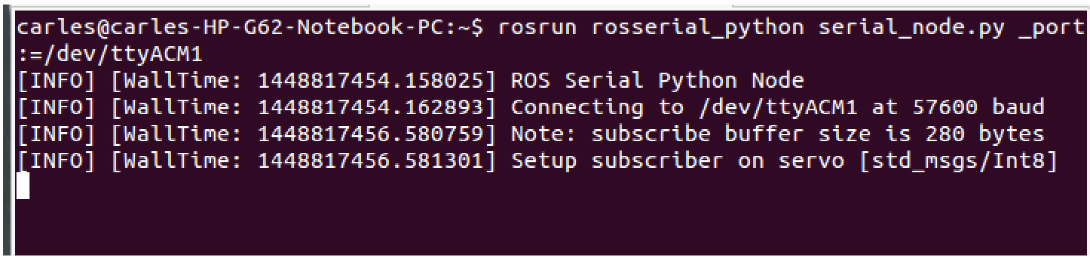
In the next page, it is explained how to control the motor:

1. First of all, the roscore has to be initialized by opening a new terminal:



```
carles@carles-HP-G62-Notebook-PC:~$ roscore
... logging to /home/carles/.ros/log/106007d4-96bd-11e5-a27a-f07bcb8abcd6/roslau
nch-carles-HP-G62-Notebook-PC-11632.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://carles-HP-G62-Notebook-PC:40192/
ros_comm version 1.11.16


SUMMARY
========

PARAMETERS
 * /rosdistro: indigo
 * /rosversion: 1.11.16

NODES

auto-starting new master
process[master]: started with pid [11644]
ROS_MASTER_URI=http://carles-HP-G62-Notebook-PC:11311/
```

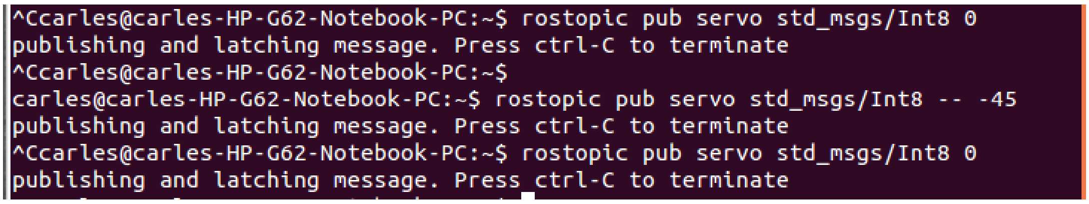
2. The USB port, where the arduino is connected, has to be defined. This also has to be done in a new terminal.
   Most of arduino have a port like /dev/ttyACM0.



```
carles@carles-HP-G62-Notebook-PC:~$ rosrun rosserial_python serial_node.py _port
:=/dev/ttyACM1
[INFO] [WallTime: 1448817454.158025] ROS Serial Python Node
[INFO] [WallTime: 1448817454.162893] Connecting to /dev/ttyACM1 at 57600 baud
[INFO] [WallTime: 1448817456.580759] Note: subscribe buffer size is 280 bytes
[INFO] [WallTime: 1448817456.581301] Setup subscriber on servo [std_msgs/Int8]
```

3. Finally, the next code line it is used to turn on the motor. The last number, in this case 8, represents the velocity of the motor. Moreover, it can also be determined the direction of the rotation. For example, -- -45 is the command to move the motor 45 rpm counterclockwise.

   rostopic pub servo std_msgs/Int XX where XX we have to write the velocity that we want.



```
^Ccarles@carles-HP-G62-Notebook-PC:~$ rostopic pub servo std_msgs/Int8 0
publishing and latching message. Press ctrl-C to terminate
^Ccarles@carles-HP-G62-Notebook-PC:~$
carles@carles-HP-G62-Notebook-PC:~$ rostopic pub servo std_msgs/Int8 -- -45
publishing and latching message. Press ctrl-C to terminate
^Ccarles@carles-HP-G62-Notebook-PC:~$ rostopic pub servo std_msgs/Int8 0
publishing and latching message. Press ctrl-C to terminate
```