Evidencia: Control



PEDRO IGNACIO IBARRA MERCADO

MORÁN GARABITO CARLOS ENRIQUE

Dinámica y Control de Robots  
9ºA

Universidad Politécnica de la Zona Metropolitana de Guadalajara.

En esta tarea se es requerido el uso de ROS, junto a algún microcontrolador cuyo objetivo es realizar un control para regular el voltaje de un potenciómetro e imprimir los datos obtenidos de su trayectoria.

Para poder realizar esta tarea se debe de conseguir los siguientes materiales:

* Microcontrolador (En este caso una Cypress).
* La instalación de ROS en un sistema compatible con este.
* La instalación o la comunicación del microcontrolador con el sistema de ROS
* Dupont
* Potenciómetro

Hay que tomar en cuenta que para realizar esta interfaz se debe de tener instalado ROS previamente y la instalación del software de tu microcontrolador, en este caso Cypress.

Una vez tomado en cuenta lo anterior se procederá con la programación del microcontrolador.

Para programar este micro, se debe de ejecutar un archivo previamente ofrecido por parte del profesor. [Aquí](RosOnAStick-master.zip)

Este software esta previamente precargado con el software para la comunicación entre Cypress y Ros.

Dentro de este archivo para cypress tiene la configuración de dicha programación para ros y un menú donde elegiremos nuestro programa a ejecutar en ros.

1.- Primero dentro de este programa abriremos la sección del lado derecho en “enconder.cpp”.

2.- Al abrir esta pestaña aparecerá el código que se modificará dependiendo de la cantidad de pasos que este gire, en este caso el motor a pasos gira un total de 540 pasos.

3.- Se modifica la comparación del dato de entrada con el dato de la cantidad de pasos ya avanzados, o los que necesita retroceder.

4.- Al ajustar estos parámetros, se debe de compilar y cargar la programación en el microcontrolador.

5.- Se enciende el sistema operativo con el contenedor de ROS, en este caso una máquina virtual.

6.- Después de haber encendido se abre una terminal y ejecutas el comando (Figura 1):

***roscore***

7.- Al correr roscore se debe de hacer un reconocimiento de los puertos pasa conocer en cual puerto se encuentra el microcontrolador (Figura 2).

**dmesg | grep tty**

8.- Una vez conectado se realiza la conexión mediante e comando (Figura 3):

***rosrun rosserial\_python serial\_node.py /dev/ttyACM0***

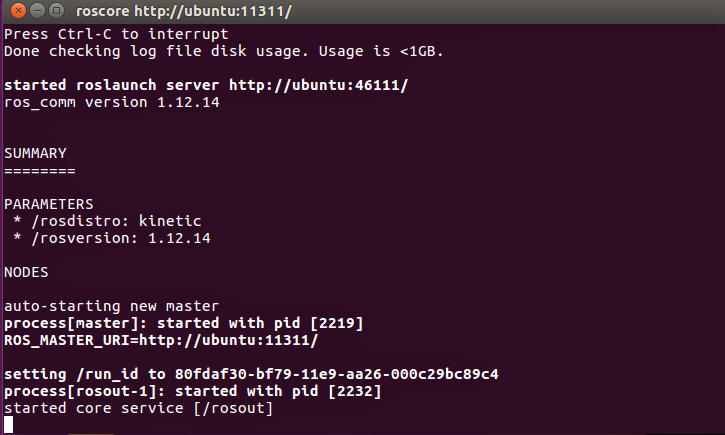
9.- Terminado esto se usa el siguiente comando para ver nuestro menú que anteriormente se programó, y se ingresa en encoder (Figura 4).

***rostopic list***

***rostopic echo /encoder***

10.- Dentro de la ventana de encoder, estará constantemente imprimiendo los datos obtenidos con los valores de voltaje del potenciómetro que empiezan y terminan desde 0v-4v (Figura 5).

Figura 1



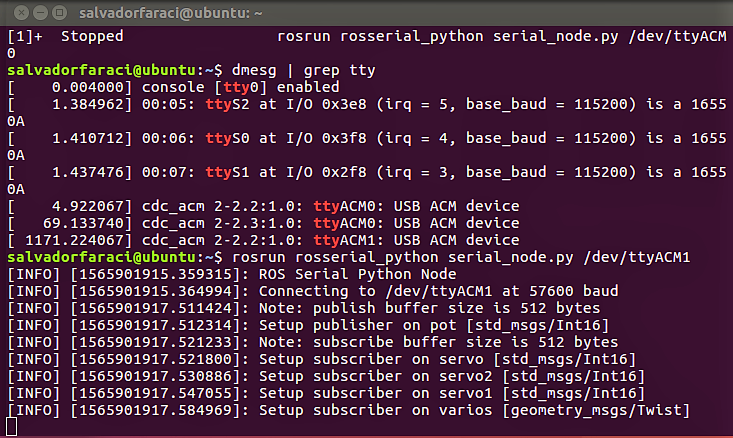
Figura 2

Figura 3

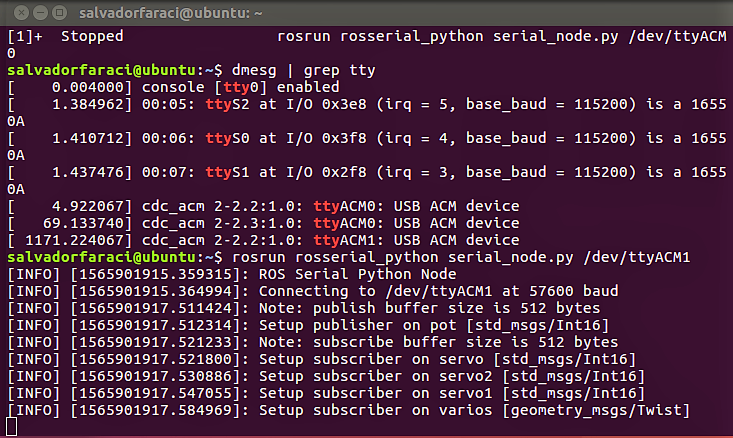


Figura 4

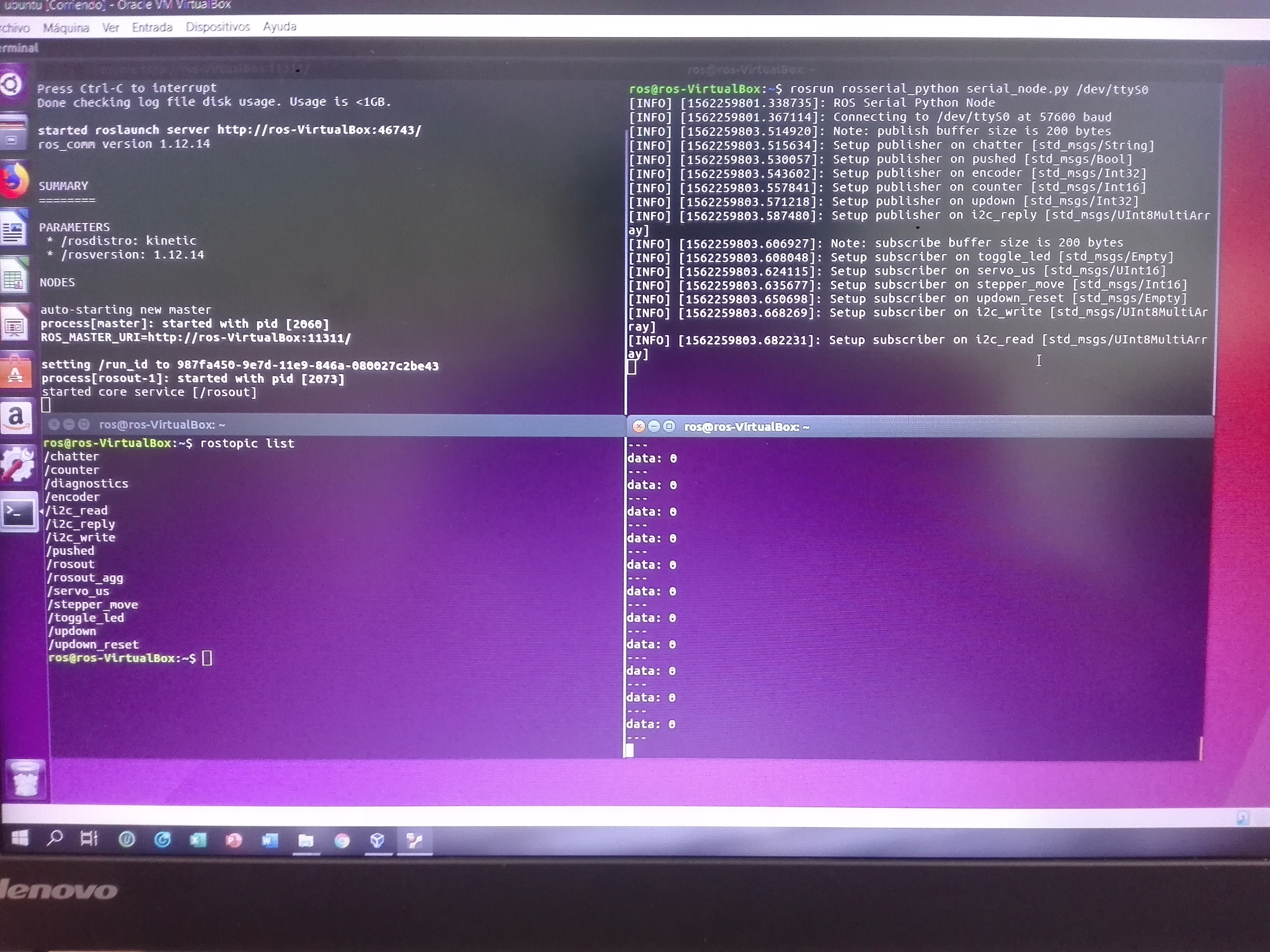
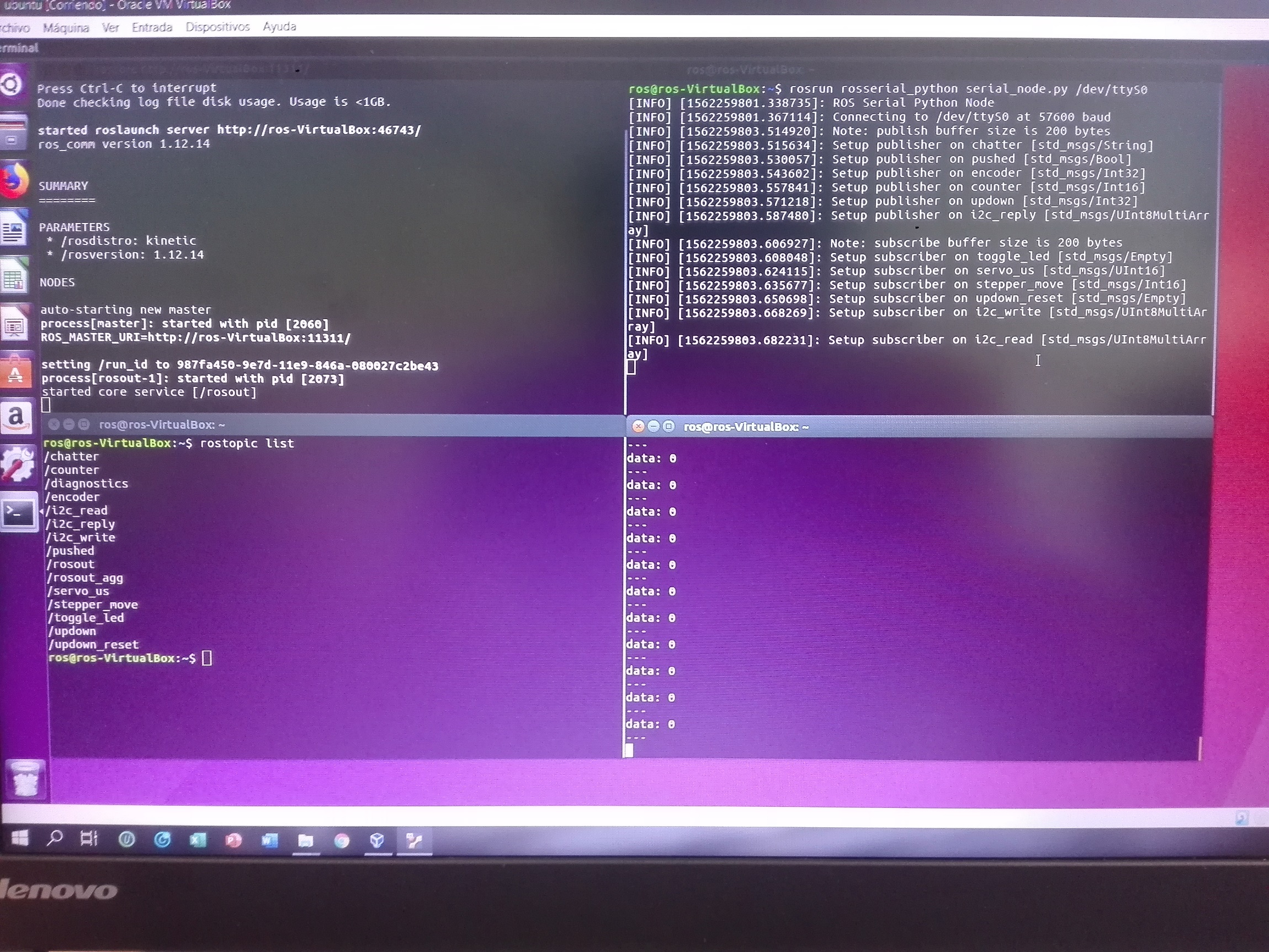


Figura 5





Conclusión:

Esta tarea a sido bastante similar a una tarea anterior, que justamente estamos usando la mayor cantidad de datos que la actividad anterior, cuya diferencia es el uso de librerías, ya que aquí lee la cantidad de voltaje usado y el otro realiza la aplicación de un control para motores a pasos.

Es práctico poder utilizar esta programación que amplia el uso de una retroalimentación para los motores tipo DC o incluso motores a pasos. Me satisface saber que se utiliza constantemente el aprendizaje adquirido y que se tiene una estrategia bien semblanteada sobre el tiempo invertido y el beneficio de este.

De cierta forma esta tarea nos puede beneficiar mucho para realizar nuestro proyecto y que espero poder aplicar lo aprendido incluso en otros tipos de proyecto.