Evidencia: Movimiento de un robot



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 con el objetivo de realizar el movimiento de este motor a pasos mediante los pasos que se desean desplazar ya sean con giros al sentido horario o inversos de este.

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
* Motor a pasos
* Driver ULN2003

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, junto a un archivo cuya explicación hace referencia a el uso de las paqueterías necesarias y distintos tipos de programas para su ejecución de pruebas. [Aquí](Tarea4.odt)

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 “stepper\_move”.

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 stepper\_move (Figura 4).

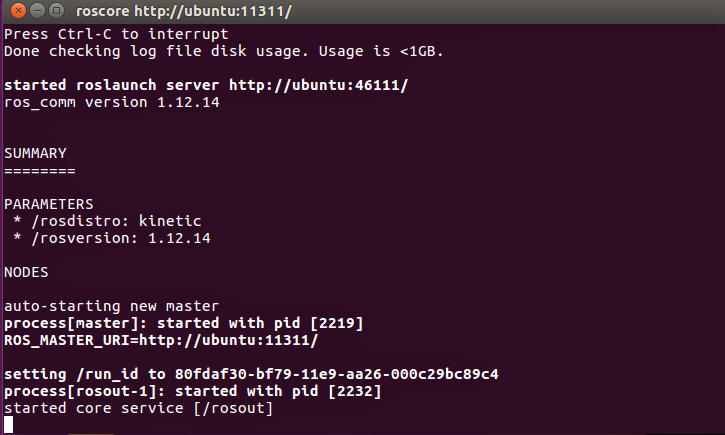
***rostopic list***

***rostopic echo /stepper\_move***

10.- Ya dentro de esta ventana el siguiente código envía la cantidad de pasos que hay que desplazarse el en motor a pasos (Figura 5):

**rostopic pub cifra std\_msgs/Int16 NUMERO\_DESEADO**

Figura 1



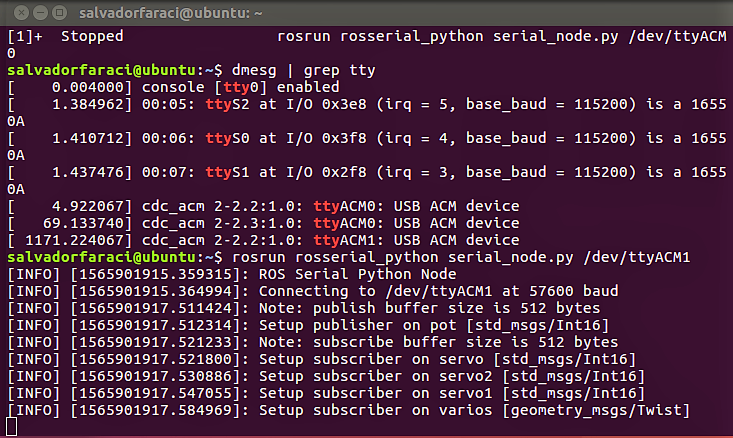
Figura 2

Figura 3

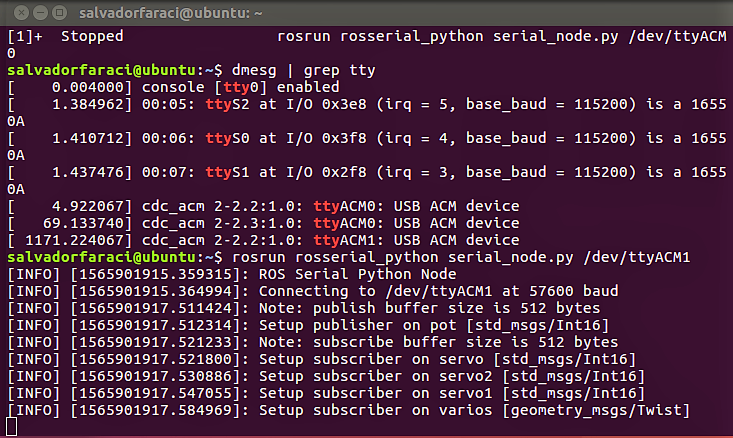


Figura 4

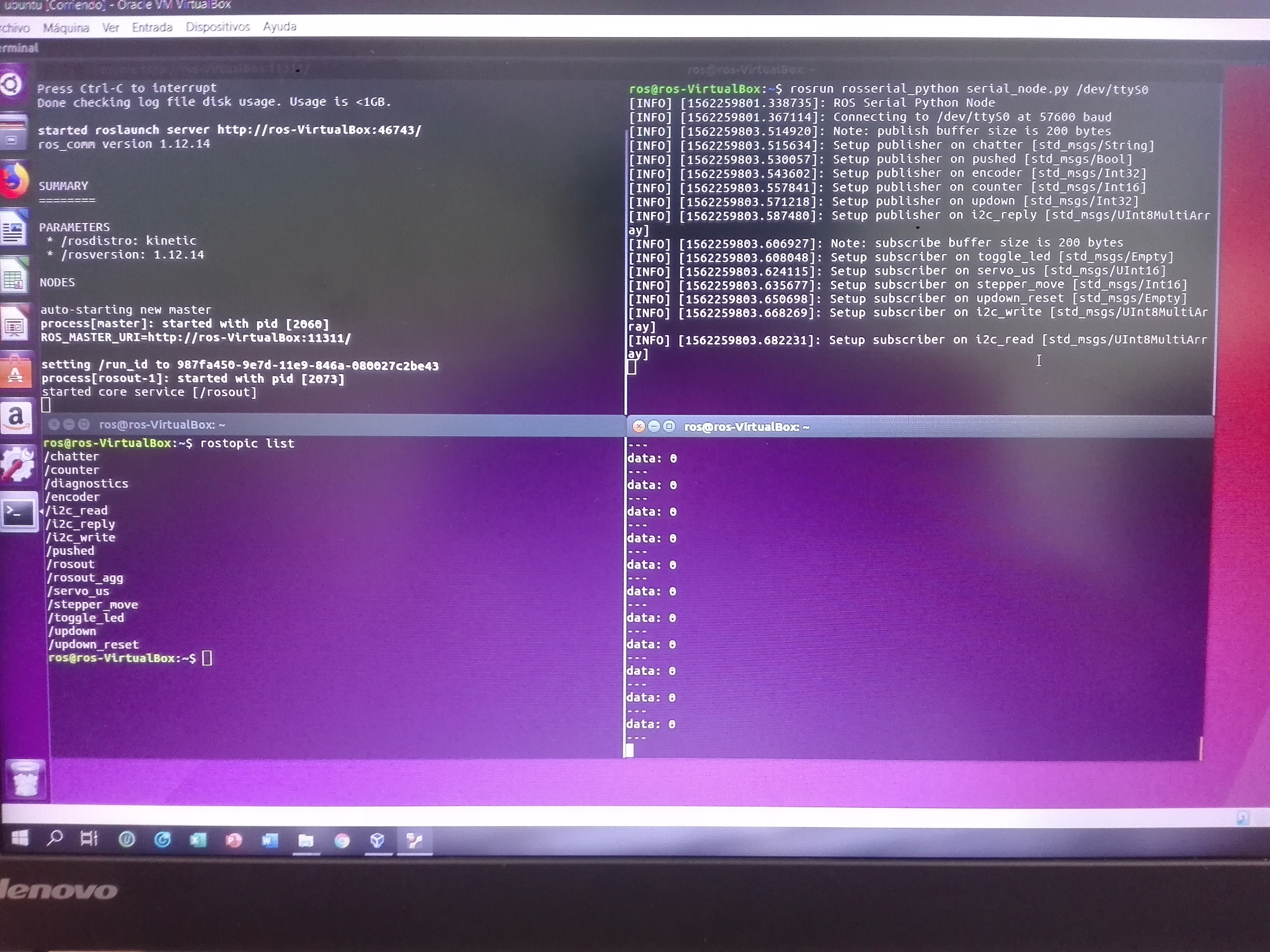
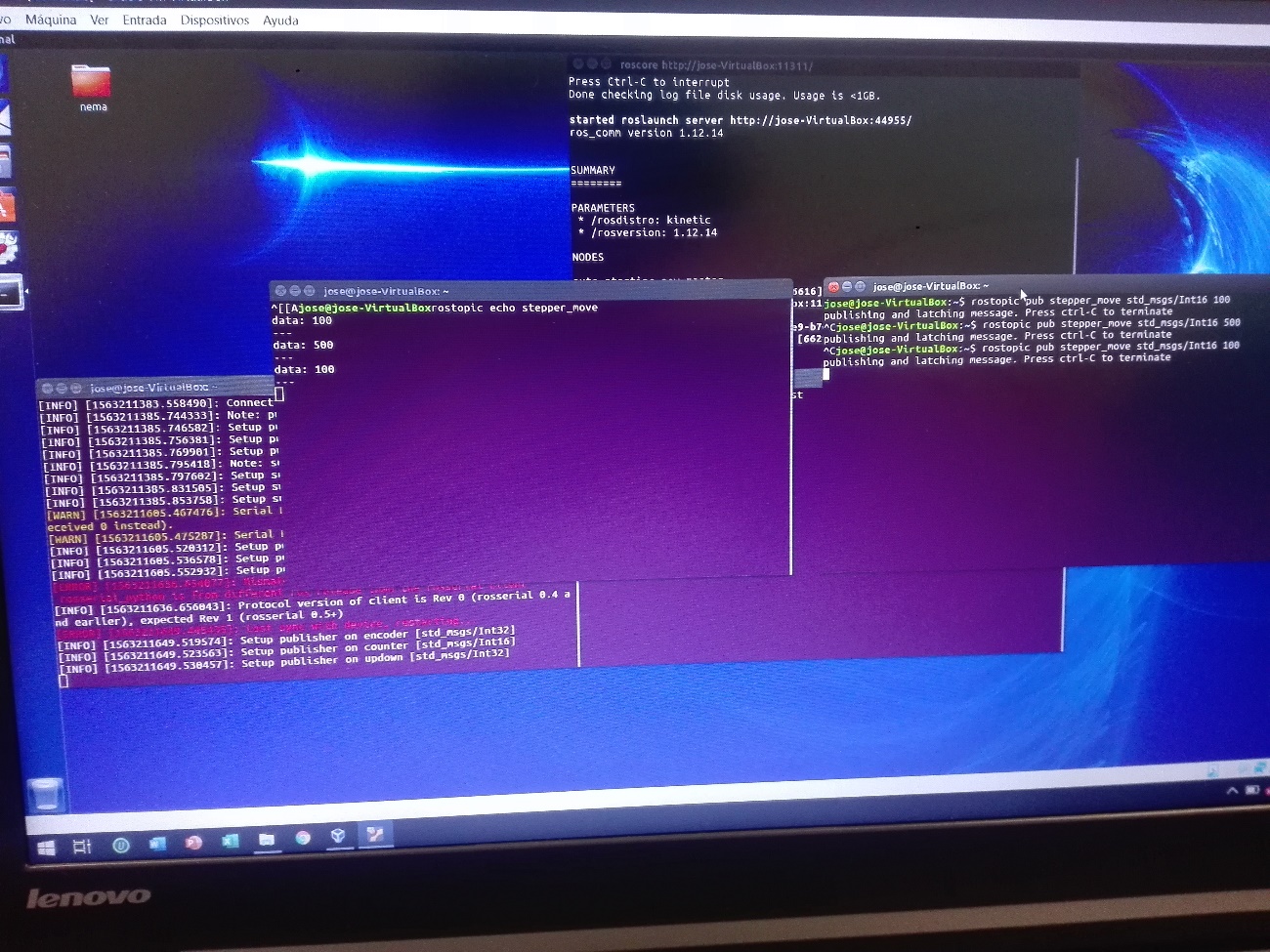
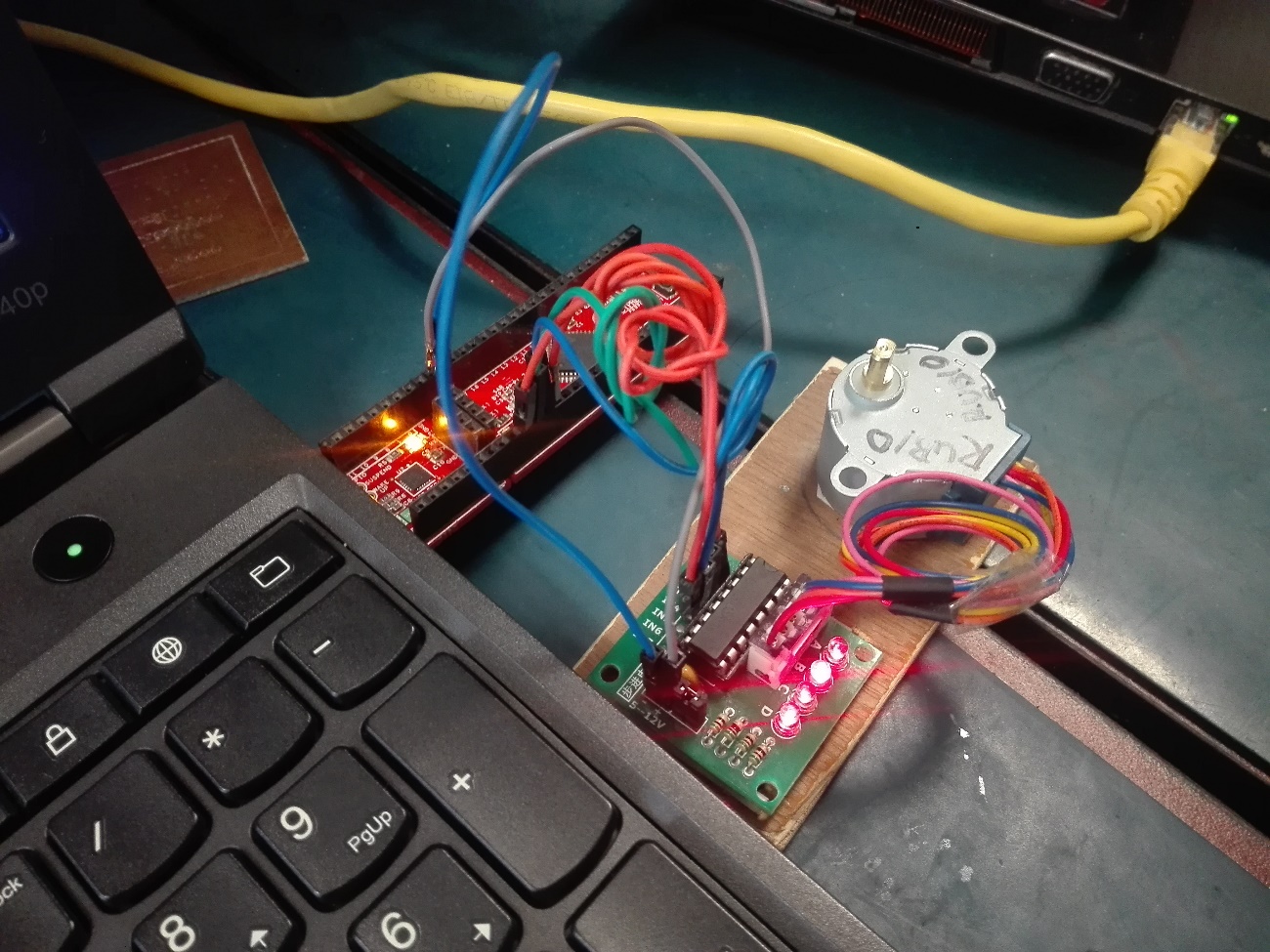


Figura 5





(Video de prueba): [Aquí](VID_20190715_122828.mp4)

Conclusión:

Esta tarea ha sido bastante dinámica para poder aprender a programar un controlador de motores a pasos, que se convierte en algo muy útil para nosotros al diseñar nuestro brazo robótico con motores a pasos y así aprender a programar los ejes necesarios.

Me a gustado el realizar esta tarea debido a que vi un buen enfoque al desarrollarla y nos ofrezca una buena opción para nuestro diseño del proyecto de material, debido a tener en práctica el conocimiento que se adquiere en el curso y que te ayude a enfocar las necesidades que requerimos de ellas.

Espero tener más tareas o prácticas en las que se desarrollen o desempeñen en estos puntos clave para mejorar nuestra perspectiva de uso del conocimiento.