

***“Script mover 3 GDL ros”***

Fonseca Camarena Jonathan

Alvarado Contreras Cesar Omar

Manzo Torres Marcos

Robles Vázquez Eduardo

Tapia Casillas Víctor Gabriel

Universidad Politécnica de la Zona Metropolitana de Guadalajara

Dinámica de Robots

**INTRODUCCIÓN:**

En esta práctica llevaremos a cabo el control de posición de un robot cnc. Al hablar de control de posición, nos referimos a que con un solo comando se debe mandar el robot a una determinada posición x, y z y con ello partir a trabajar de un punto de origen determinado como puede llamarse coloquialmente “home”.

El control de posición lo realizaremos con sensores de final de carrera, esto quiere decir que al introducir un comando, los motores comienzan un retroceso hasta que el sensor de carrera los interrumpe, de esta manera tenemos la posición. Los sensores fueron colocados de manera específica en los lugares donde se tendrás que tomar las coordenadas de posición.

**MARCO TEÓRICO:**

El corazón del sistema de comunicaciones serie es la UART, acrónimo de Universal Asynchronous Receiver-Transmitter.  Es un chip cuya misión principal es convertir los datos recibidos del bus del PC en formato paralelo, a un formato serie que será utilizado en la transmisión hacia el exterior.  También realiza el proceso contrario: transformar los datos serie recibidos del exterior en un formato paralelo entendible por el bus.

La UART es un dispositivo programable en el que pueden establecerse las condiciones que se utilizarán para la transmisión (velocidad, paridad, longitud y bits de parada).  En los primeros PCs, eran circuitos integrados 8250 de National Semiconductor (un chip de 40 patillas DIP -Dual-In-Line-), pero se han ido utilizando otros a lo largo de la evolución del PC.

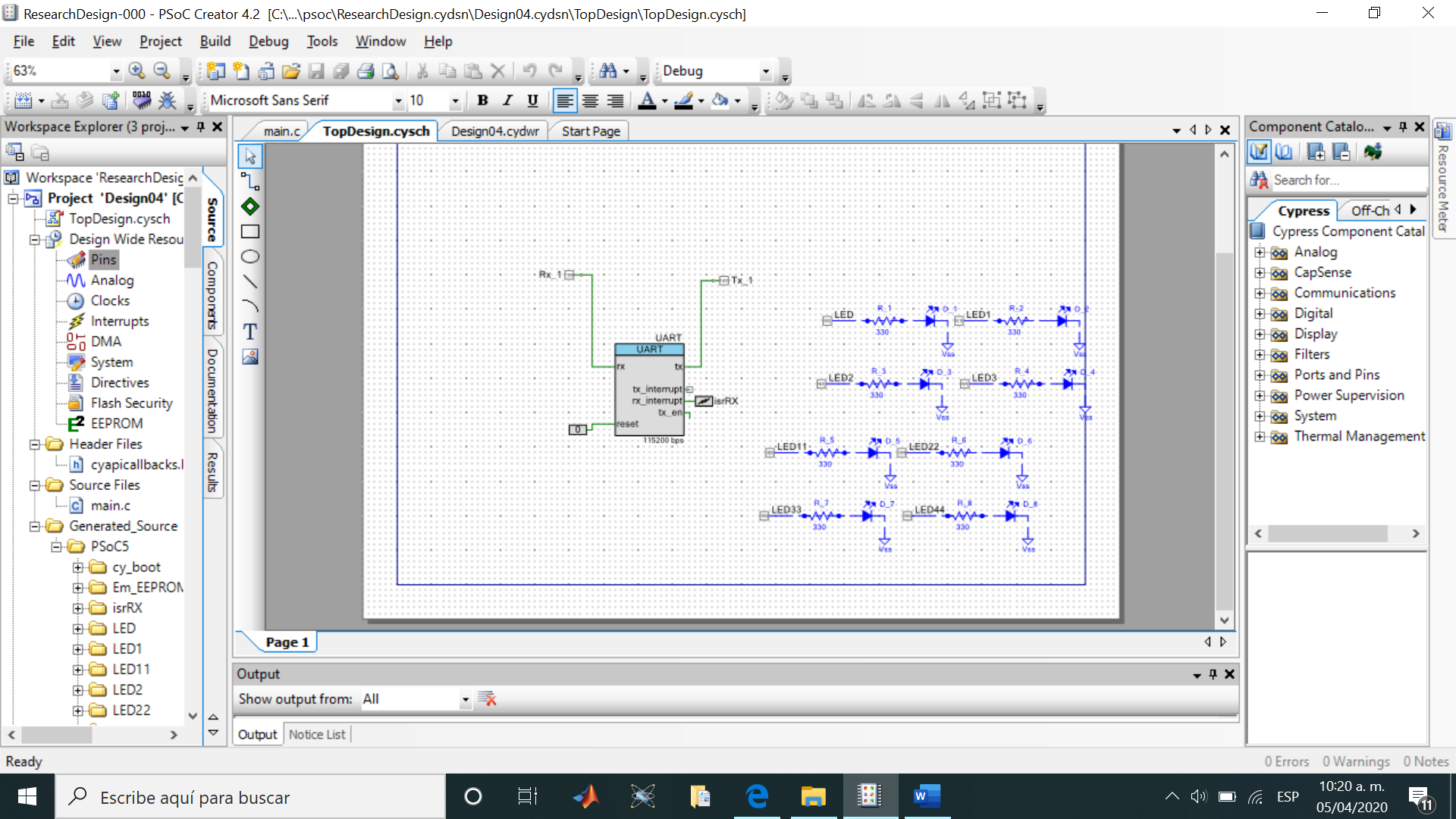
El UART o USART es el puerto serie hardware que todos los microcontroladores tienen al menos uno y la comunicación serie es la base de casi cualquiera de las comunicaciones de los microcontroladores.

El “Bus Universal en Serie”, en inglés: Universal Serial Bus más conocido por la sigla USB, es un bus estándar industrial que define los cables, conectores y protocolos usados en un bus para conectar, comunicar y proveer de alimentación eléctrica entre computadores, periféricos y dispositivos electrónicos.

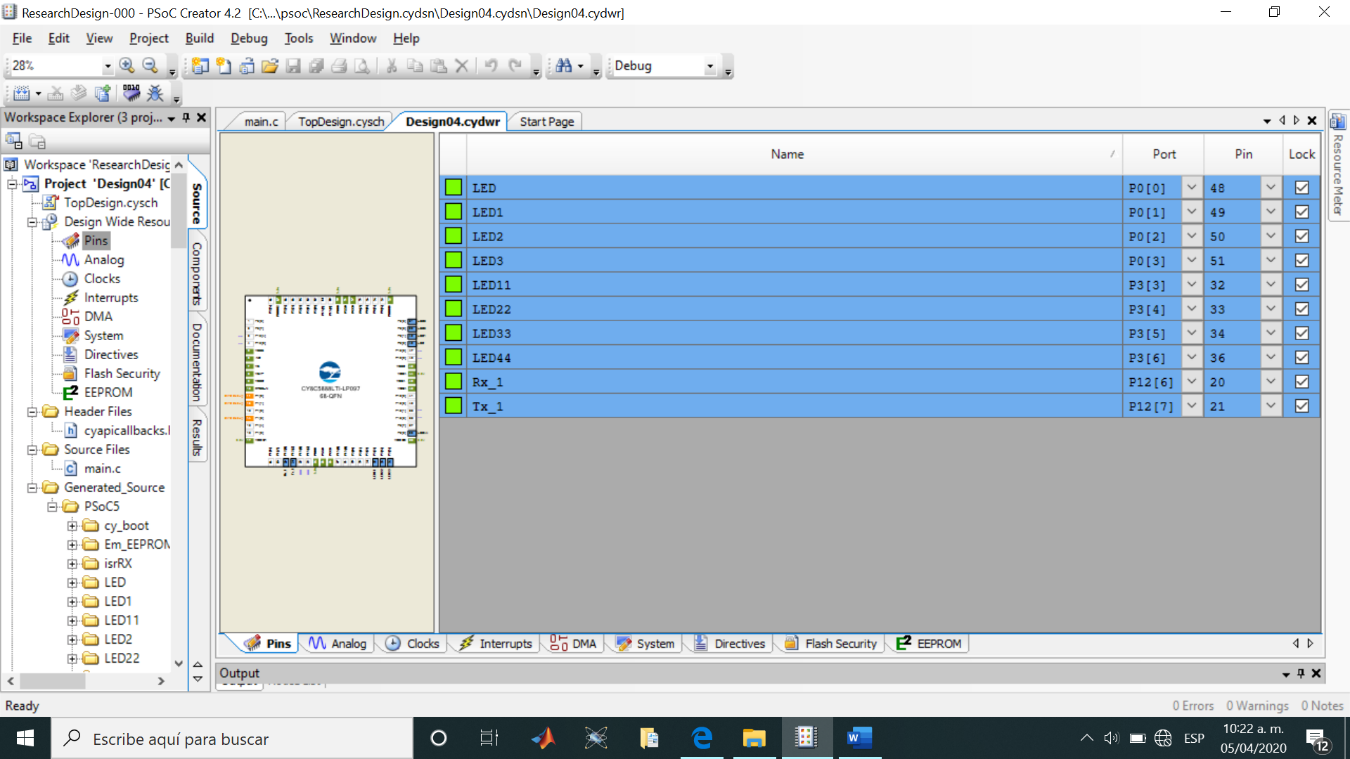
Su desarrollo partió de un grupo de empresas del sector que buscaban unificar la forma de conectar periféricos a sus equipos, por aquella época poco compatibles entre si, entre las que estaban Intel, Microsoft, IBM, Compaq, DEC, NEC y Nortel. La primera especificación completa 1.0 se publicó en 1996, pero en 1998 con la especificación 1.1 comenzó a usarse de forma masiva.

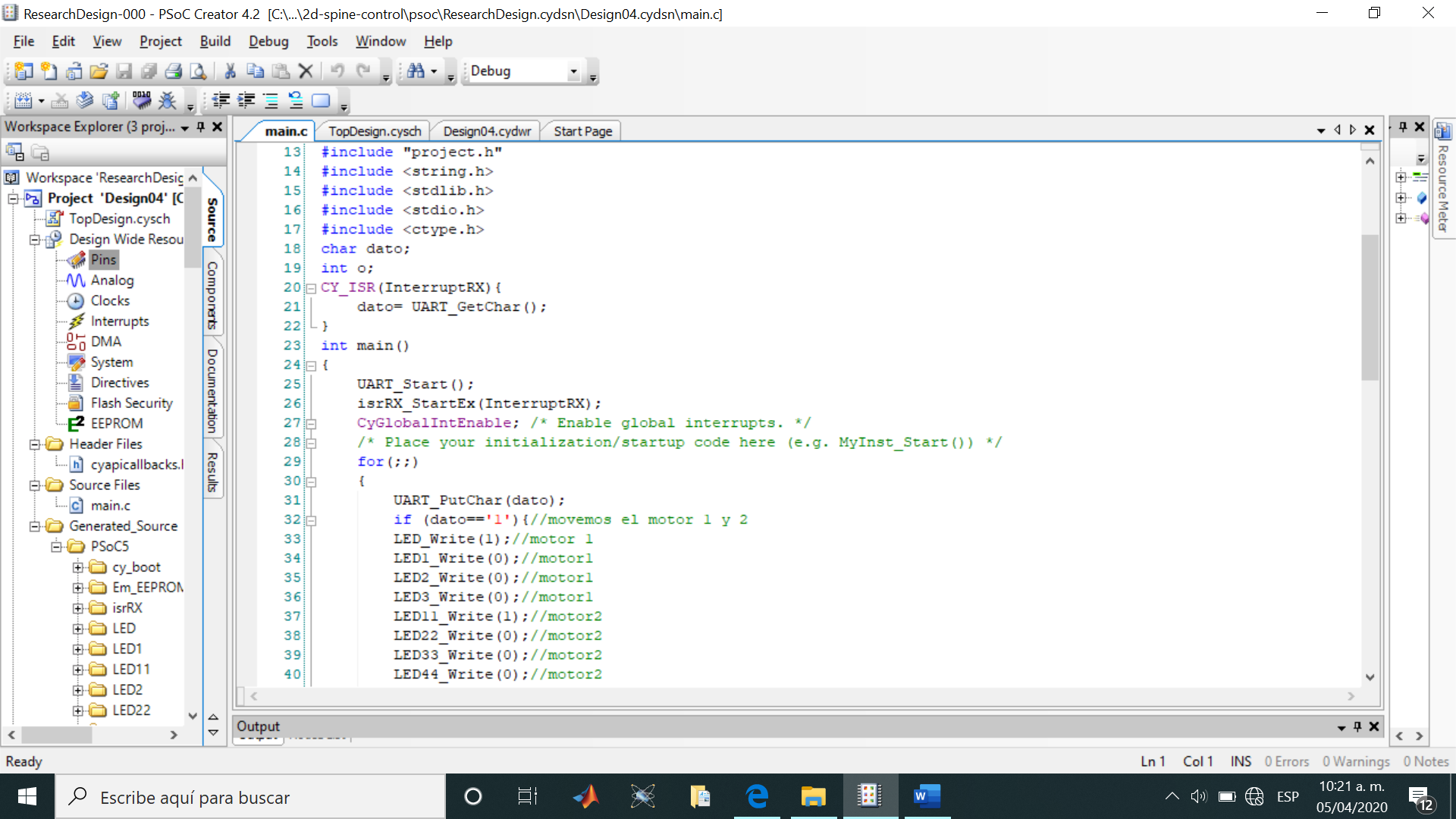
**DESARROLLO:**

Como se ha venido trabajando, el control lo realizaremos con la tarjeta psoc 5lp. Básicamente lo que necesitamos es declarar 8 led’s, en este caso, cuatro leds por cada motor ya que el driver que estaremos manejando trabaja con cuatro entradas y cuatro salidas al motor.



Observamos la programación, básicamente decimos que al obtener un valor “1” por el puerto serial, el motor se activará y comenzará el funcionamiento.



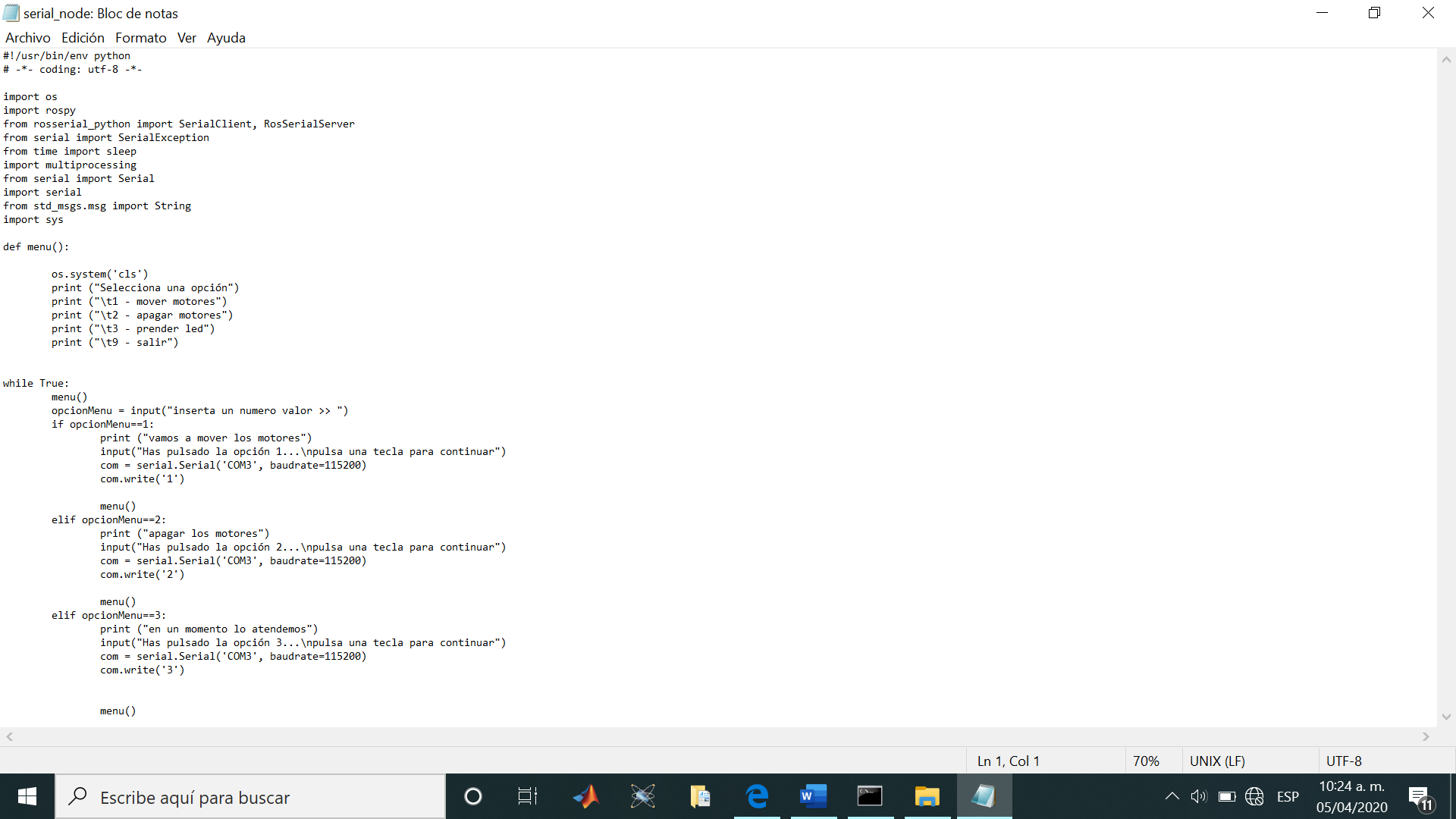


Una vez observado el código del controlador, pasamos al código que estaremos ejecutando en conjunto con ROS.

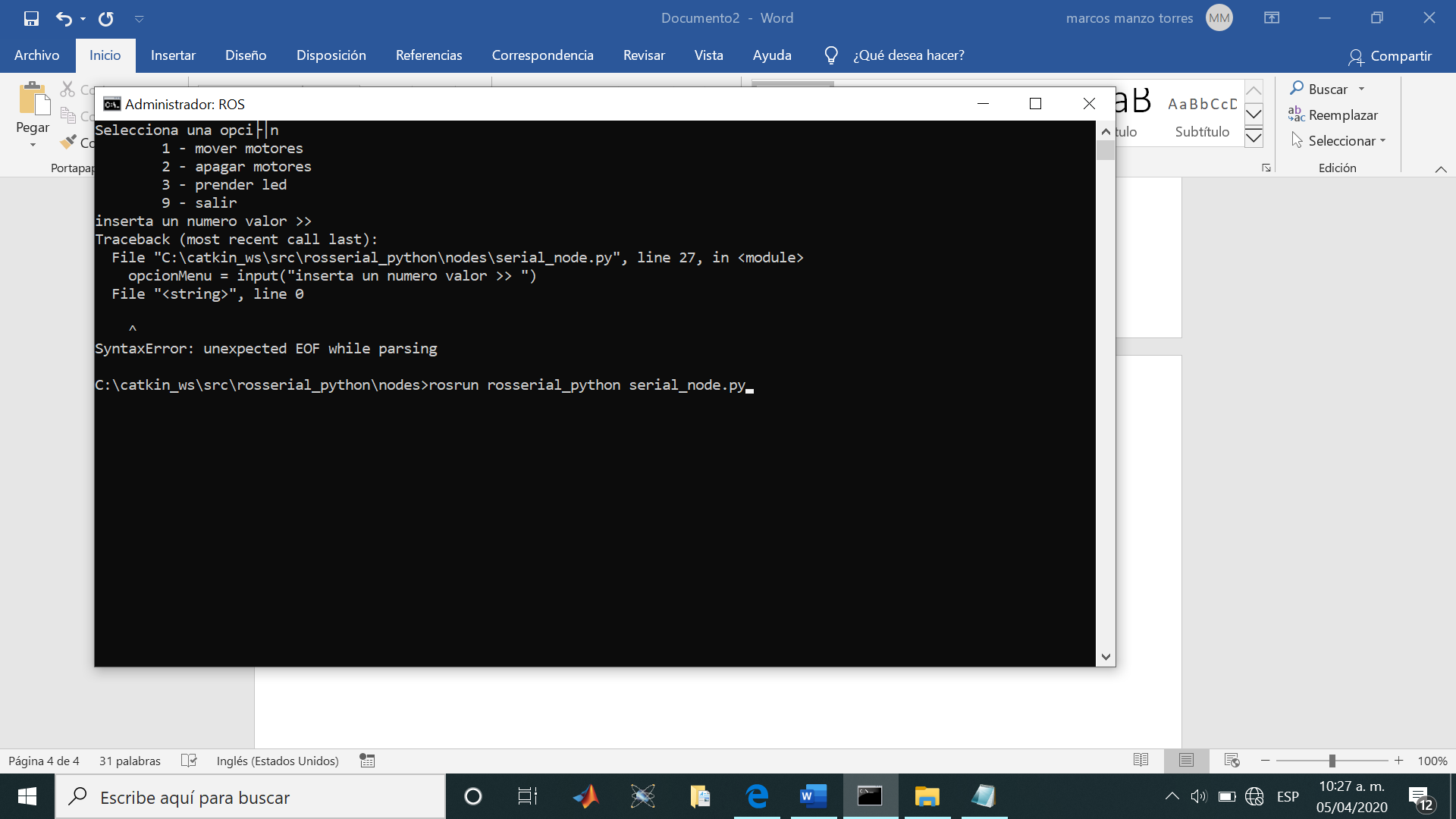
Básicamente lo que hacemos, es enviar el carácter configurado al puerto serial para que de esta manera el puerto serial active los motores.

Se realizó un menú, para tener de una manera más amigable la programación para el puerto serial

El código fue realizado en Python.

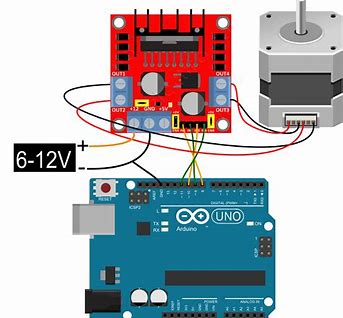


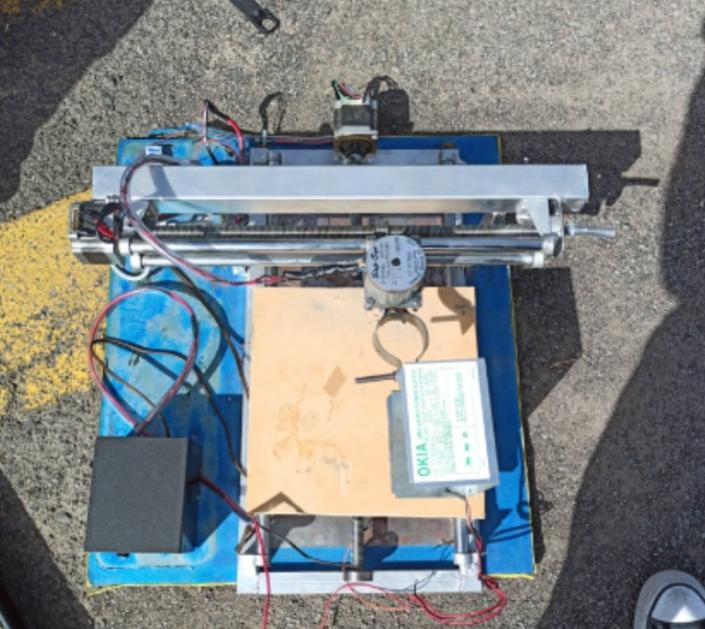
Para ejecutar el código, nos ubicamos en la carpeta donde se ubica y las subcarpetas. La carpeta fue realizada de la forma catkin\_ws <http://wiki.ros.org/catkin>



**CONTROL DE LOS MOTORES**

El control de los motores, lo realizamos con el driver l298, que maneja un voltaje externo de 12 volts y que recibe cuatro entradas del controlador para tener cuatro entradas de salida que van a las bobinas del motor a pasos.





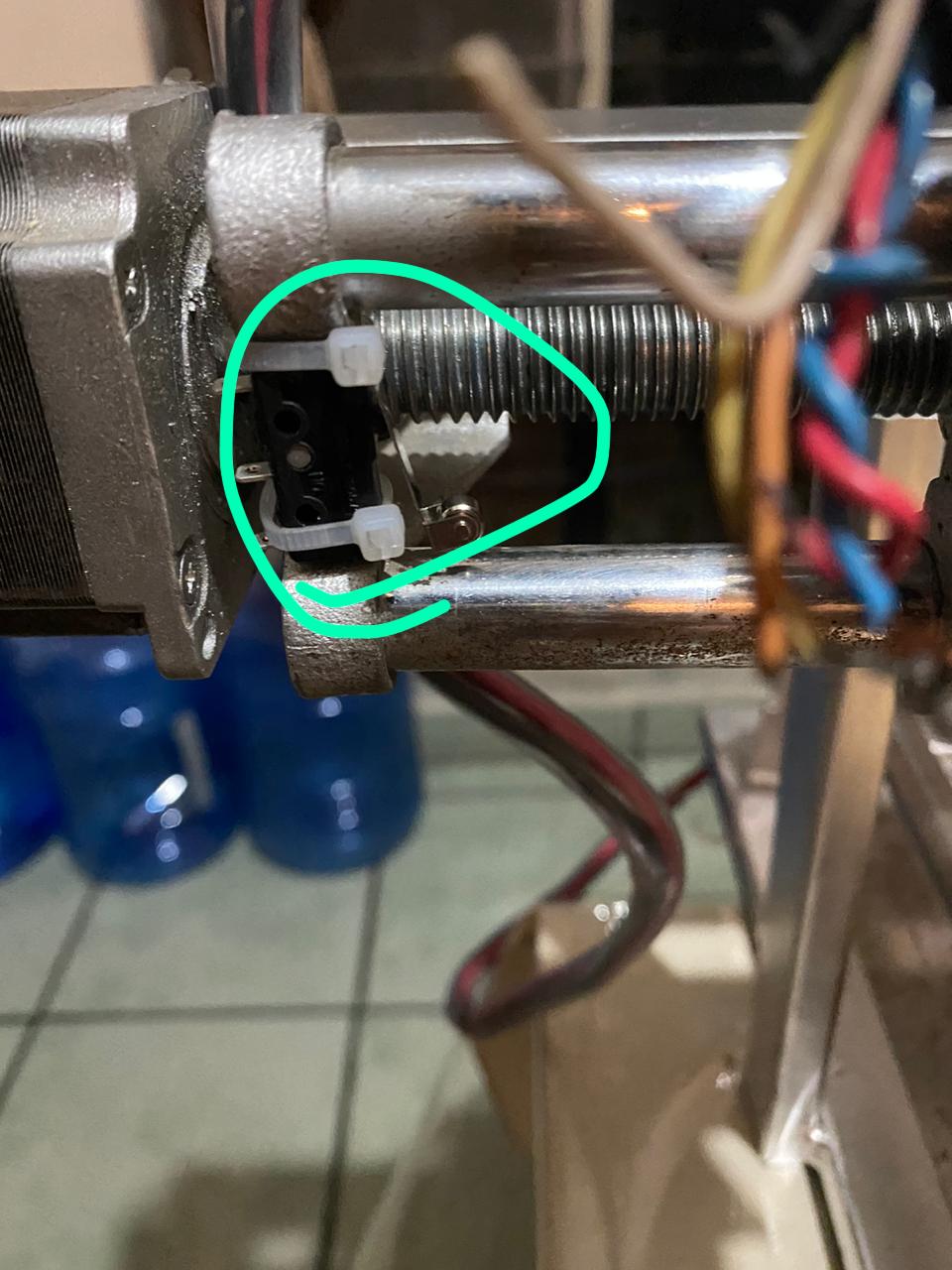
Los motores usados son nema23

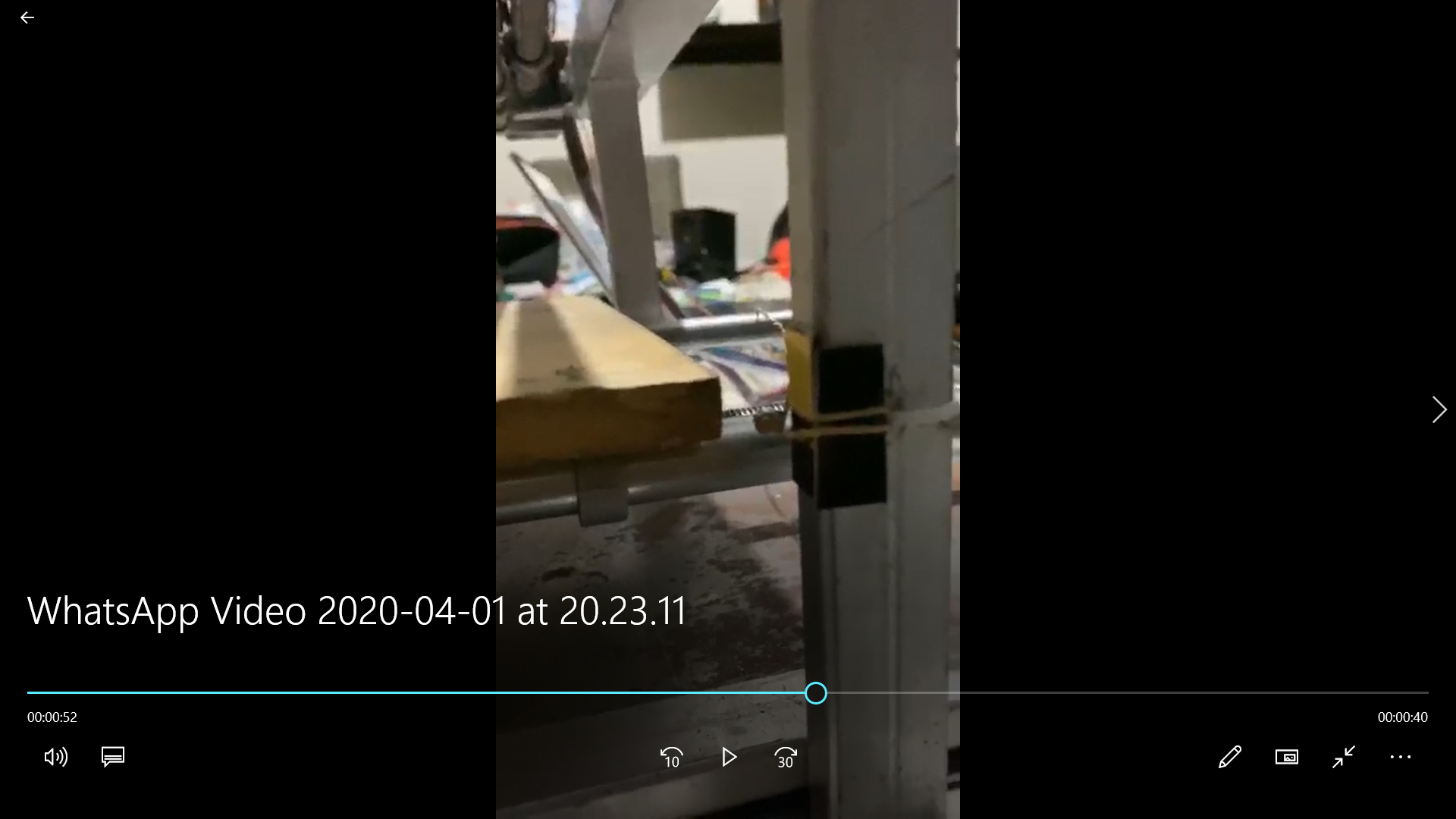
**CONTROL DE POSICIÓN:**

El control de posición lo realizamos con sensores de final de carrera, que al oprimirse interrumpen la acción, en este caso, frenan el paso del motor

De esta manera, al llegar a la ubicación de los tres sensores, el robot partirá su trabajo de dicho lugar y podrá trabajar con una orientación determinada y coordenadas específicas teniendo un home en sus filas de trabajo

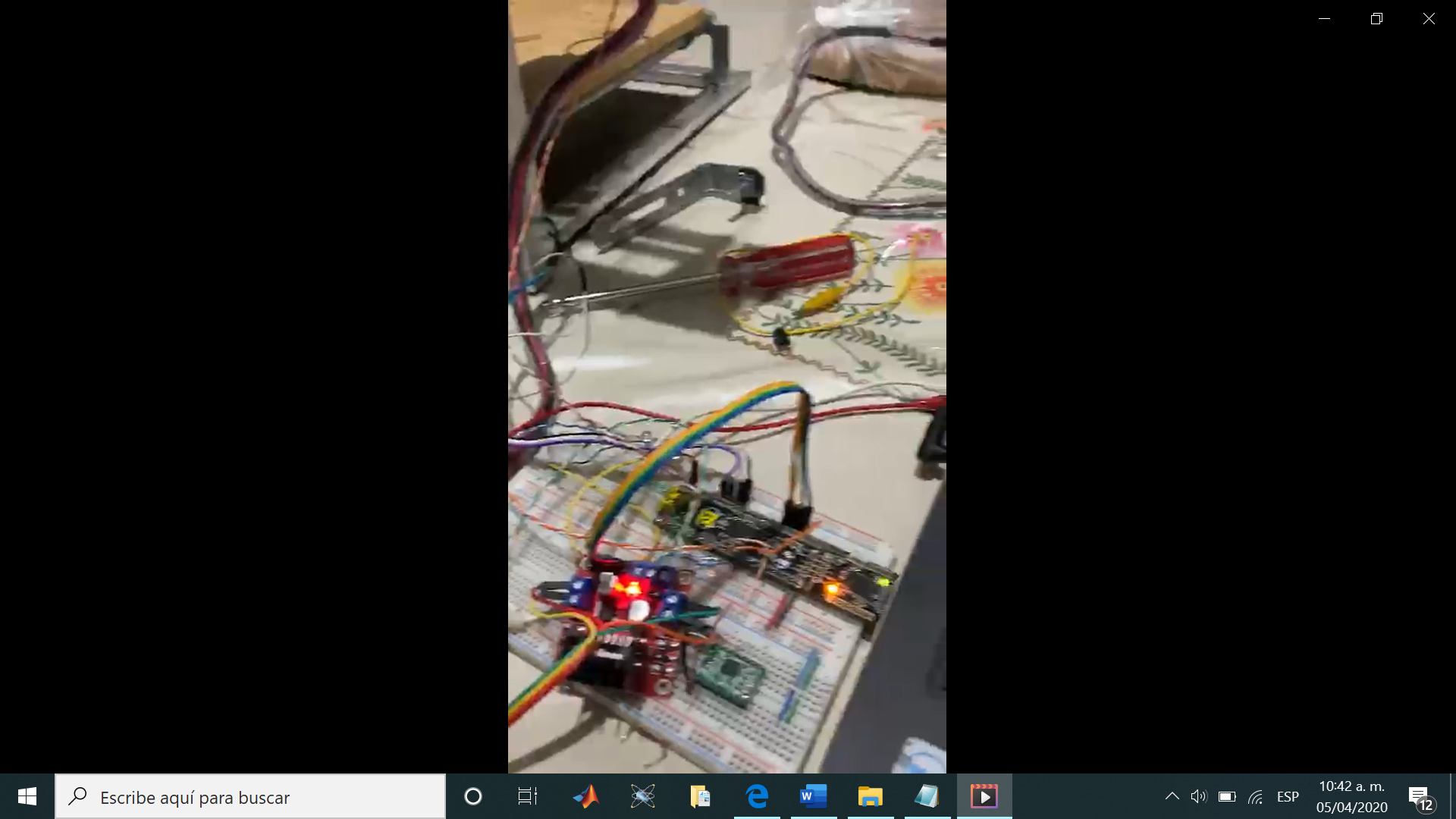


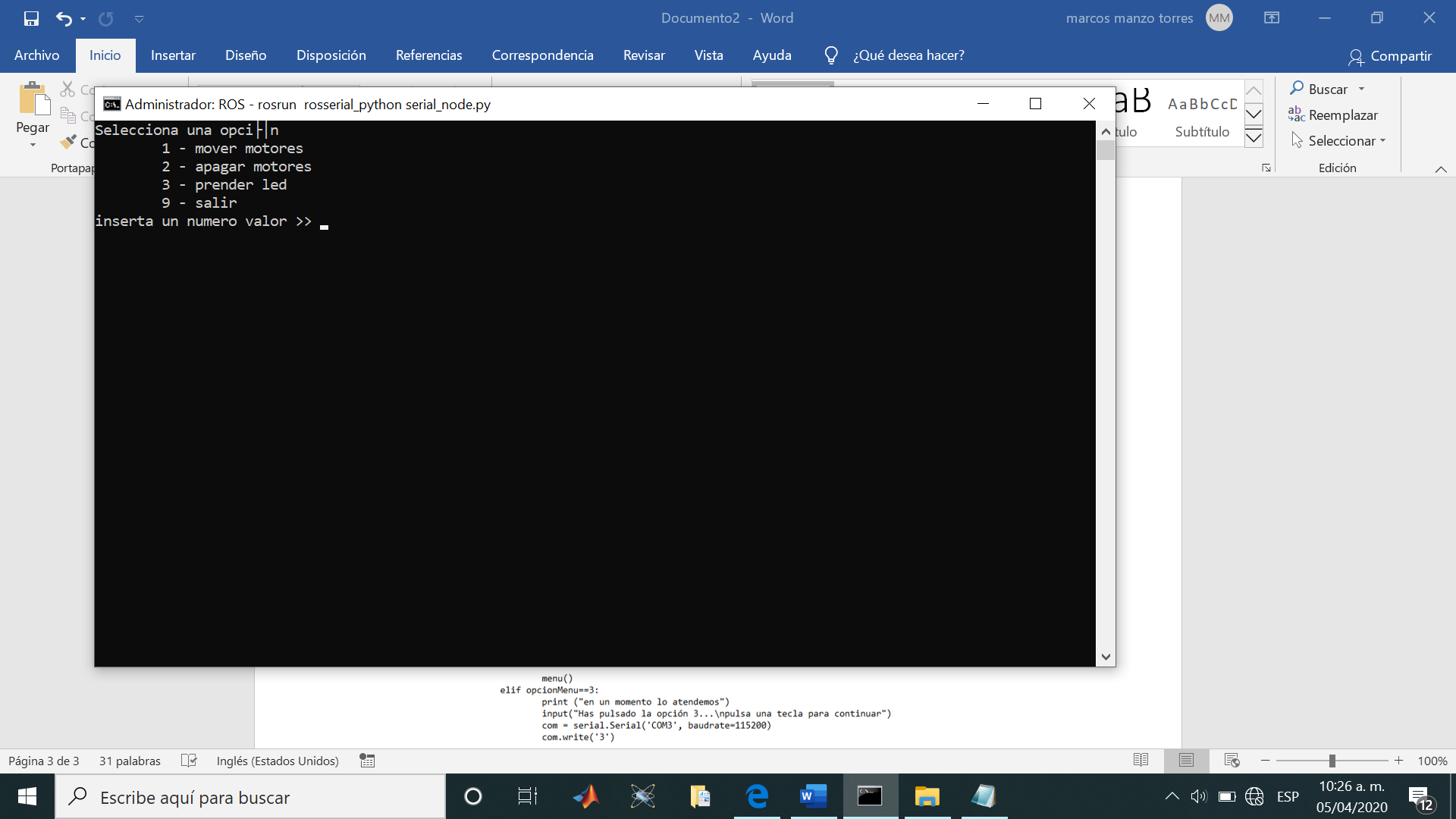




**CONEXIONES DEL ROBOT CON LA PSOC**

Observamos las conexiones de la PSoC con los drivers del motor, de esta manera ya tenemos todo listo para comenzar a trabajar con nuestro robot cnc



Una vez que tenemos todo lo anterior, podemos lanzar el código con ros y comenzar a trabajar con el robot cnc partiendo de una posición específica, ubicado el robot en un espacio determinado

**VIDEO DEL FUNCIONAMIENTO**

[**https://youtu.be/qOYjMD9ASb8**](https://youtu.be/qOYjMD9ASb8)

**CONCLUSIÓN:**

Esta práctica fue de un gran aprendizaje, debido a que aprendemos a realizar el control preciso del robot, de esta manera llegamos a realizar el funcionamiento continuo del robot.

Partiendo de la idea de un control, por ejemplo, laser GRBL, tenemos lo que sería los puntos claves como mover a la derecha, mover a la izquierda y un punto de home de donde podemos partir todos los trabajos.

Es clave antes de realizar algo físico, saber que es lo que esperamos a la salida, de esta manera ya tenemos lo que viene siendo el funcionamiento correcto de acuerdo a lo que esperamos.

Realizando el control de posición, es el primer punto para comenzar a trabajar ya de una manera más directa con el control. Comenzando con ello, podemos comenzar a trabajar en todas las posiciones de manera pronunciada.