Laboratorio 06: Control de un motor de pasos

Por:

Ian Gabriel Cañas Fernández, 1092228

.

.

# Control de un motor de pasos

Reutilizando el funcionamiento del teclado matricial para controlar un motor de pasos.

# Objetivos:

* Diseñar un sistema digital que permita el control de un motor de pasos.

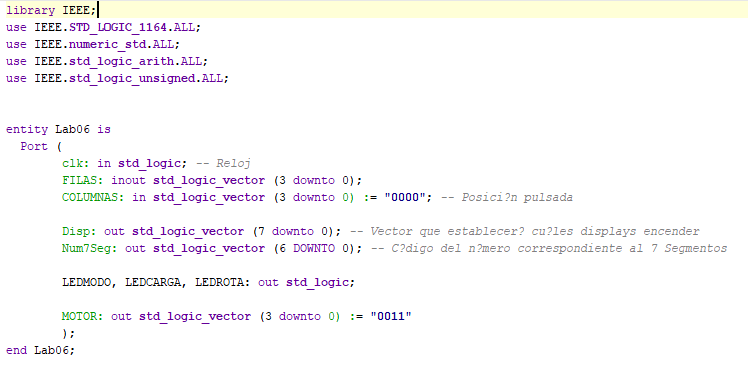
# Procedimiento:

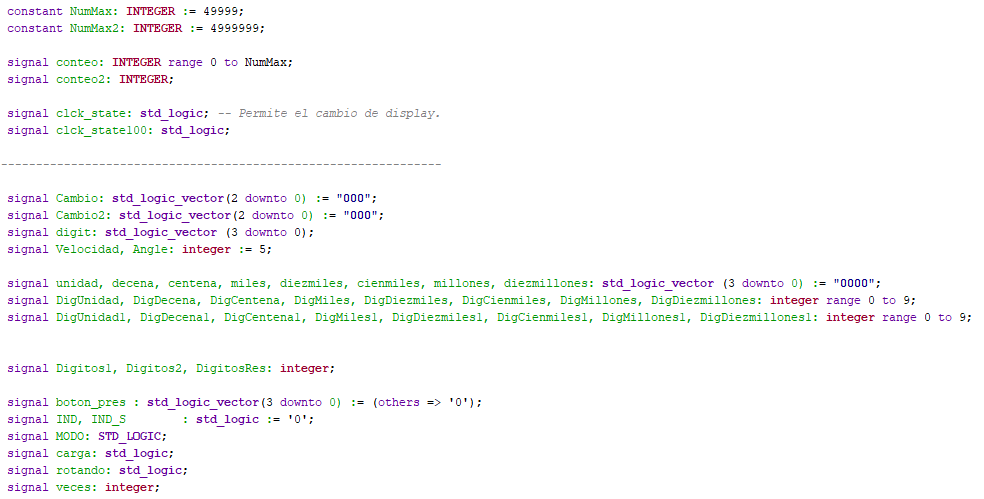
En el presente laboratorio estamos reutilizando tanto el funcionamiento del teclado matricial 4x4 como el funcionamiento del display y todo lo que hemos visto con anterioridad. En este preciso caso estamos utilizando el teclado matricial para introducir los valores de velocidad y ángulo de giro.

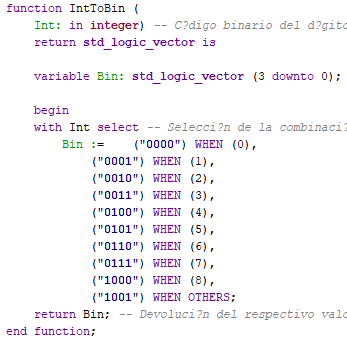
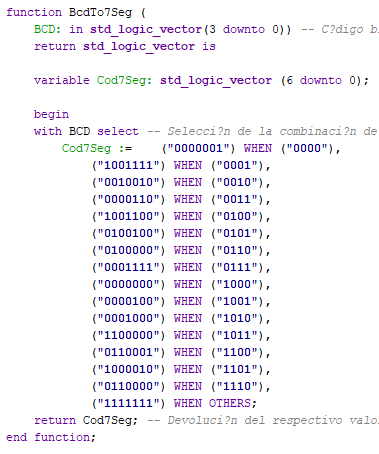
Al ser el mismo código del teclado utilizado en el laboratorio 05, se obviará la explicación de este.

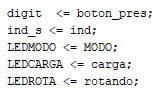
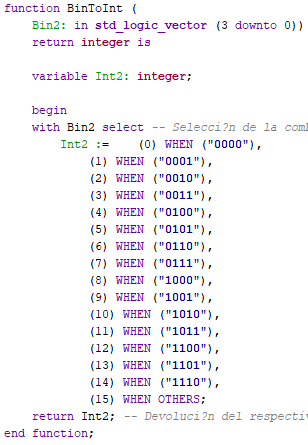
## El control del motor de pasos

Empezamos generando los puertos, funciones y señales necesarias para el laboratorio:

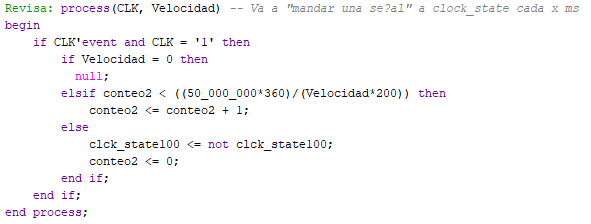




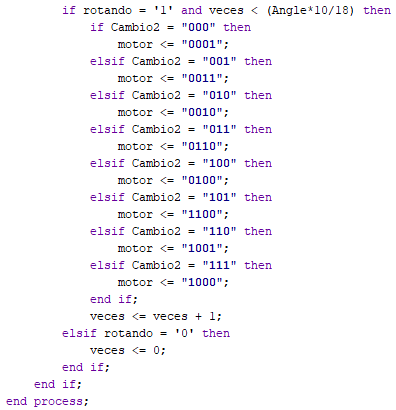
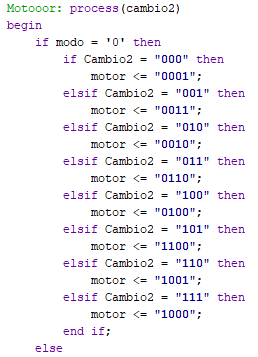




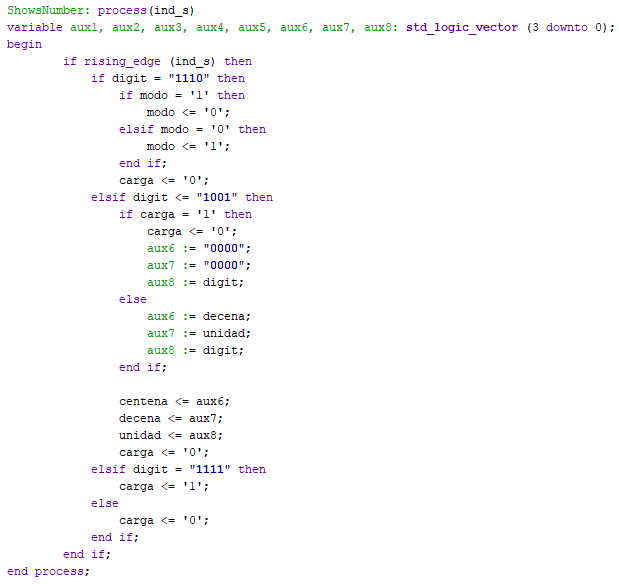
Generamos, entonces, una función que genera una frecuencia que depende de la velocidad introducida para no utilizar velocidades predeterminadas, mediante el uso de la siguiente ecuación:



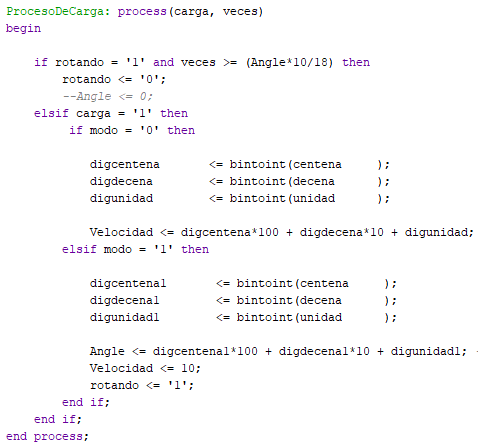
No es muy preciso el resultado, pues se está trabajando con enteros, pero se mantiene en un rango aceptable. Hemos de considerar que se está utilizando frecuencia media porque ha de suceder 2X veces para darse un paso del motor, ya que cada paso se llama cuando clck\_state100 llega vuelve a su valor ‘1’.



Luego creamos el proceso que identifica el botón siendo pulsado:



Además, creamos el proceso que recibirá un aviso del proceso anterior de si se cargará el número actual o de un cambio de modo:

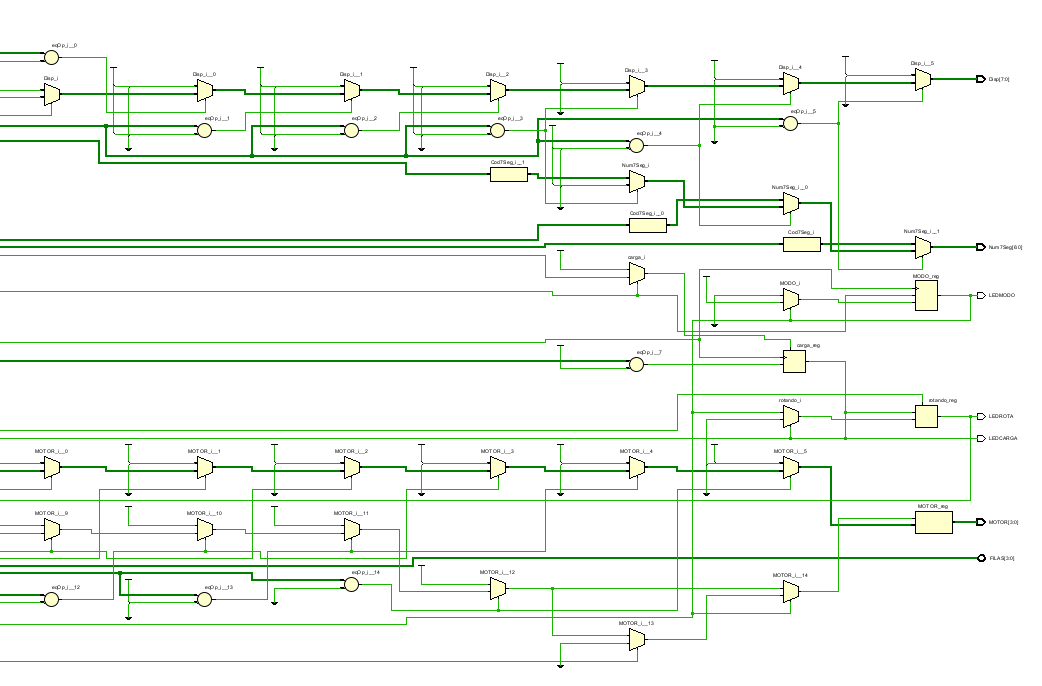
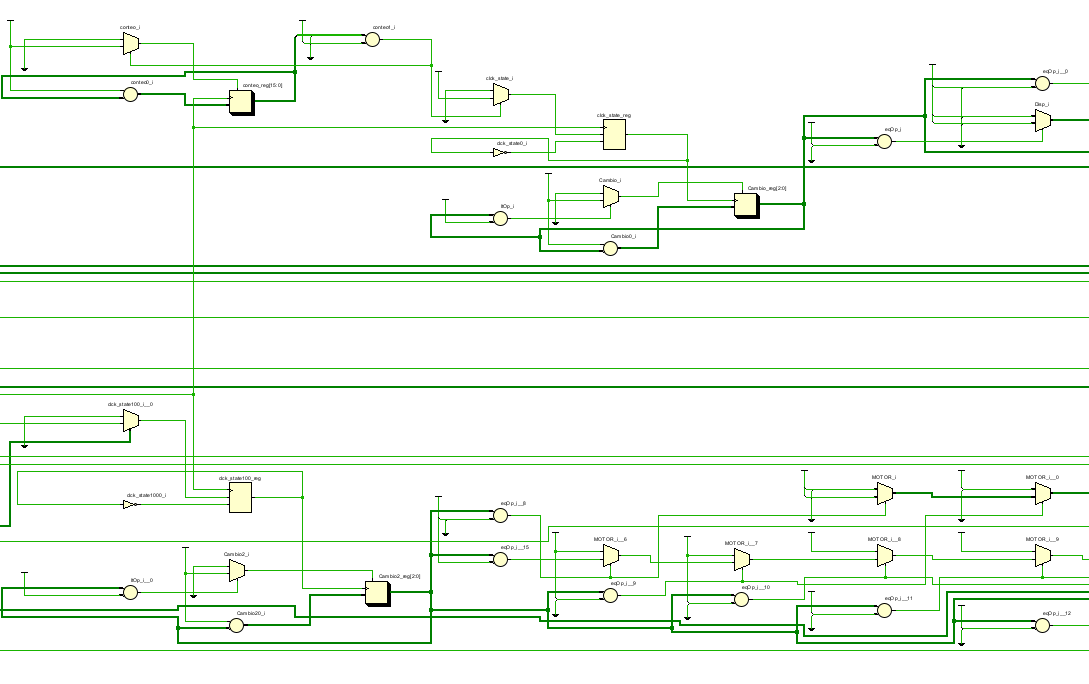
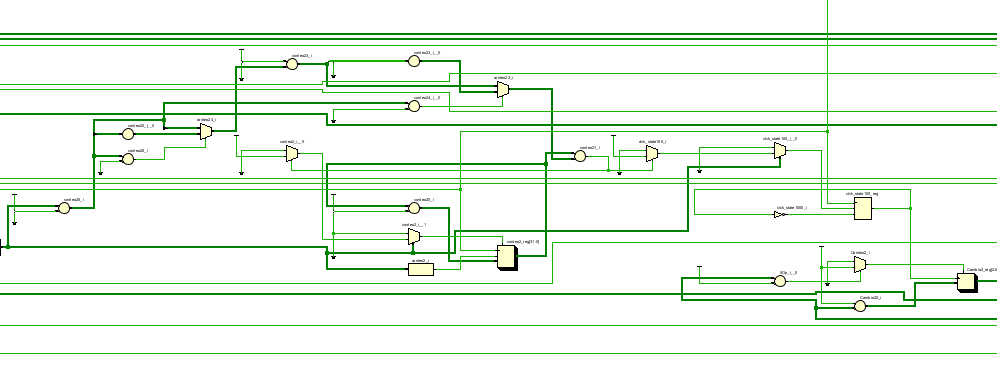
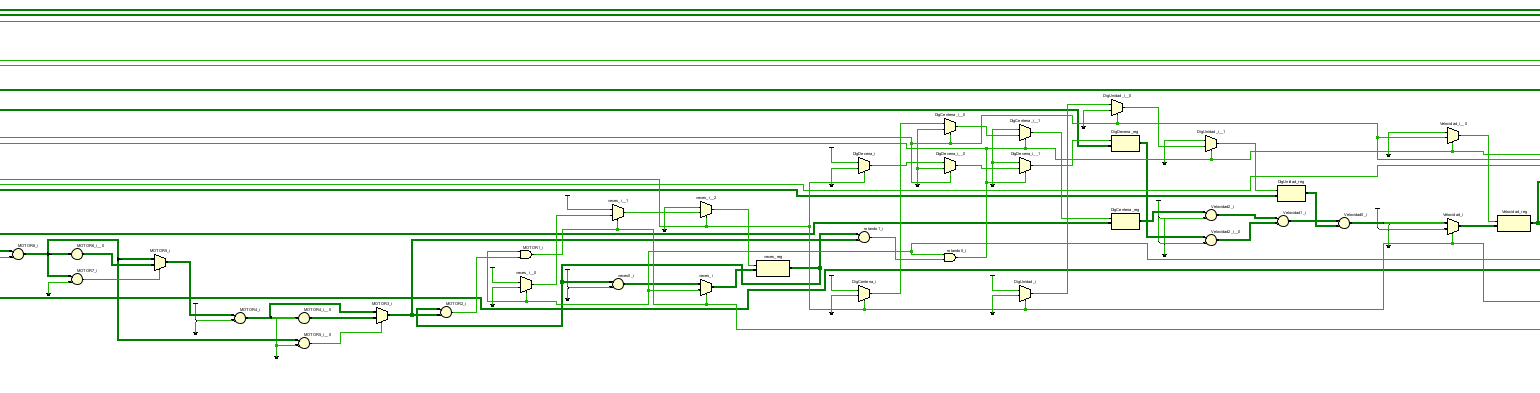
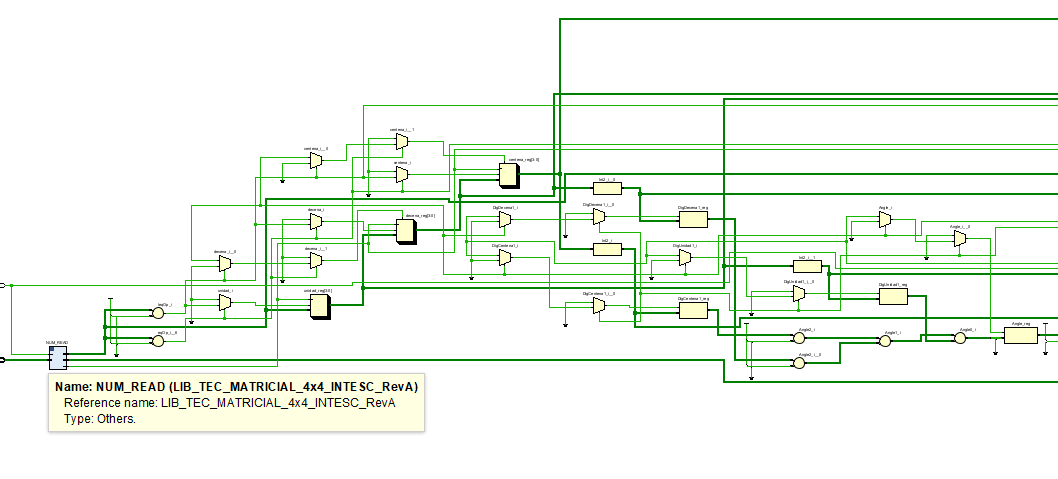


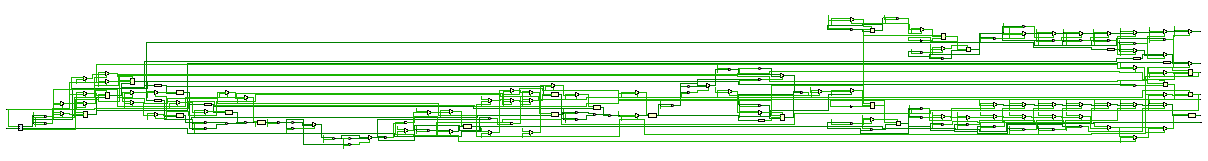
Hemos de notar que aquí, igualmente se cambia el status de rotando a ‘1’ o a ‘0’, esta señal permite la rotación del motor en el proceso donde se dan los pasos en el caso de un ángulo específico. ¿Cómo? Pues en ese proceso se hace un conteo de cantidad de pasos totales a dar, y en caso de dar un paso más al contado inicialmente, en este proceso se comprueba si llegamos a dicha cantidad y la señal rotando se pasa a ‘0’, impidiendo así la reactivación de un paso a menos que intencionalmente se vuelva a activar.

# Resultados:

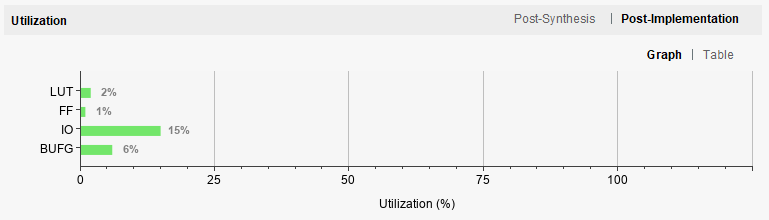
Hemos obtenido una reacción positiva por parte del motor. Este varía su velocidad con una certeza sorprendente, mientras mayor el valor introducido, mayor la velocidad, tal y como se estimaba. El mismo caso con el ángulo, el motor inicia su recorrido y se detiene en un ángulo cercano al especificado.

Además, se obtiene el siguiente diseño con la misma complejidad al anterior:

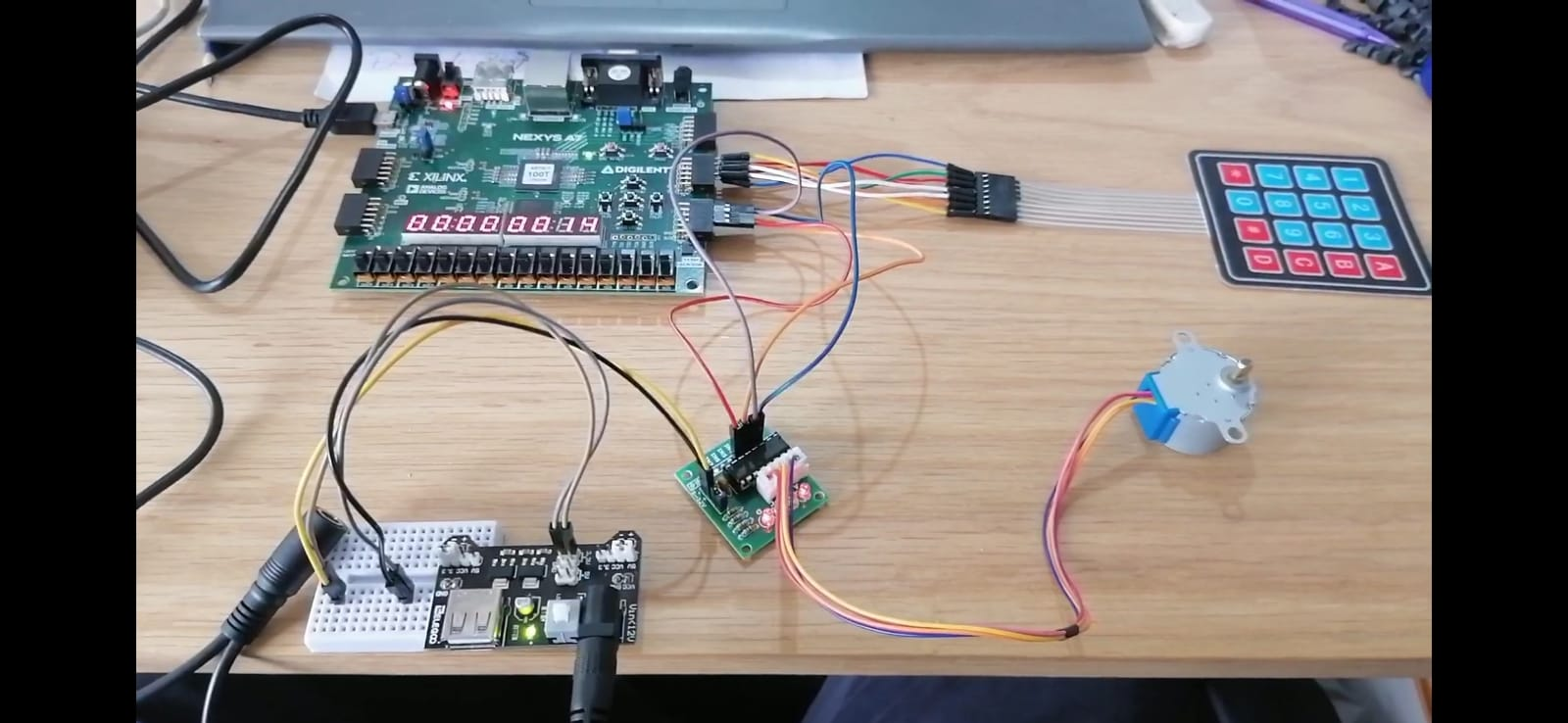




A pesar de la ”complejidad” presentada, e programa estima el siguiente uso de recursos:



Nos encontramos ante la siguiente configuración física. Hemos de considerar que el motor stepper ha de ser alimentado mediante una fuente externa, pues no es aconsejable alimentar el motor con la misma tarjeta de desarrollo, pues esta no posee tal potencia. Además, siempre se aconseja el uso de un driver para el motor para evitar daños.



# Análisis:

Se pudo comprender el uso del motor de pasos (stepper) y su funcionamiento, así como su aprovechamiento físico. Se observa que el funcionamiento correcto de este motor depende de una señal que se le sea enviada dígase con un patrón cíclico o secuencial. Incluso, mediante el mismo método, su dirección puede ser fácilmente cambiada de sentido horario a antihorario, o viceversa.

En el proceso de diseño, pudimos recalcar sin temor a equivocación, que no es ni fácil ni útil el asignar un valor a una señal en procesos diferentes, aunque estos no se activen al mismo tiempo.

Mediante la aplicación física se pudo observar que el motor tiene una velocidad literal limitada a menos de 90 grados por segundo, esto se debe a que, como es necesario que cruce corriente en ciertos puntos en tiempos distintos para activar un imán, no alcanza el tiempo para manifestarse este efecto físico.

Finalmente, hemos logrado controlar un stepper tal y como se esperaba con la distinción entre velocidad y cantidad de vueltas.