



**UNCUYO**  
UNIVERSIDAD  
NACIONAL DE CUYO



**FACULTAD  
DE INGENIERÍA**

# Robot Graffiti

Proyecto Final de la Cátedra Microcontroladores y Electrónica de  
Potencia

Alumno: JOSE, Nahuel

Profesor Titular: IRIARTE, Eduardo

# 0.Índice

## [0.Índice](#)

## [1.Introducción](#)

## [2.Descripción del sistema](#)

### [2.0.Geometría](#)

### [2.1.Control y comunicación](#)

### [2.2.Consideraciones constructivas](#)

#### [2.2.0.Dimensionamiento de la zona de trabajo](#)

#### [2.2.1.Cálculo de velocidades](#)

#### [2.2.2.Coordinación de los motores](#)

## [3.Funcionamiento del sistema](#)

### [3.0.Máquina de Estados](#)

#### [3.0.0.Master](#)

#### [3.0.1.Slave](#)

### [3.1.Desarrollo de la comunicación y la secuencia de estados](#)

#### [3.1.0.Estado inicial - Desenergizado](#)

#### [3.1.1.Energizado](#)

#### [3.1.2.Referenciando](#)

##### [3.1.2.0.Referenciamiento del esclavo 1](#)

##### [3.1.2.1.Estado intermedio](#)

##### [3.1.2.2.Referenciamiento del esclavo 2](#)

##### [3.1.2.3.Estado Final](#)

#### [3.1.3.Referenciado](#)

#### [3.1.4.Movimiento sin dibujar](#)

#### [3.1.5.Esperando punto final](#)

#### [3.1.6.Dibujando](#)

##### [3.1.6.0.Preparándose para dibujar](#)

##### [3.1.6.1.Dibujando](#)

##### [3.1.6.2.Fin del dibujo](#)

#### [3.1.7.Solicitud de estado](#)

## [4.Elección y Cotización de Elementos](#)

### [4.0.Motores](#)

### [4.1.Microcontroladores](#)

### [4.2.Piezas 3D](#)

#### [4.2.0.Mordaza para dispositivo de escritura](#)

#### [4.2.1.Poleas](#)

#### [4.2.2.Ensamble](#)

### [4.3.PCB](#)

## [5.Repositorio GitHub](#)

## [6.Trabajo a futuro](#)

# 1.Introducción

La propuesta realizada para el trabajo final de la cátedra fue el control de la posición de un dispositivo de impresión o de escritura en una superficie vertical, entendiéndose un aerosol sobre un muro o un marcador en una pizarra.

El objetivo es diseñar y programar un sistema que recibiendo el punto inicial y final de una línea, en coordenadas cartesianas, sea capaz de ejecutarla sobre el muro, pizarra o lienzo.

El artefacto se concibió de la siguiente manera, un dispositivo suspendido desde los vértices superiores del área de trabajo unido por “cuerdas” a poleas solidarias a los ejes de los motores paso a paso que enrrollarían y desenrrollarían dichas cuerdas controladamente para llevar el dispositivo a la posición correspondiente. Por practicidad y para proteger los componentes (principalmente los motores), se consideró preferible colocar en dichos vértices simplemente unas poleas que cambien la dirección de las cuerdas y los motores se ubicarían al nivel del suelo.

Para lograr un buen resultado los dos puntos importantes a tener en cuenta son:

- El control de la posición del dispositivo que realiza el dibujo y
- El desplazamiento conjunto y coordinado entre los dos motores de este dispositivo.

Es un problema de control de posición en dos dimensiones, es decir, de un sistema de dos grados de libertad, con una cinemática particular, no solamente por la disposición vertical, sino también porque los desplazamientos no son independientes y la necesidad de coordinación de los desplazamientos.

## 2.Descripción del sistema

### 2.0.Geometría

El área de trabajo posible consta de un cuadrado de 1768mm aproximadamente de lado (más adelante se justificará esta medida), lo que significa un área de trabajo de 3,1 m<sup>2</sup> aproximadamente.

Como se puede inferir de la Imagen , las coordenadas impuestas para la expresión de los puntos inicial y final de las líneas a dibujar tienen su origen en la esquina superior izquierda de este cuadrado (“O” en la Imagen ), con el eje x horizontal con la dirección positiva hacia la derecha y el eje y vertical con la dirección positiva hacia el suelo.

Entonces, este sistema debe convertir estas coordenadas (x,y) a las coordenadas (d<sub>1</sub>,d<sub>2</sub>) que representan las distancias desde las esquinas y a su estas distancias a pasos de los motores (m<sub>1</sub>,m<sub>2</sub>), de la siguiente manera:

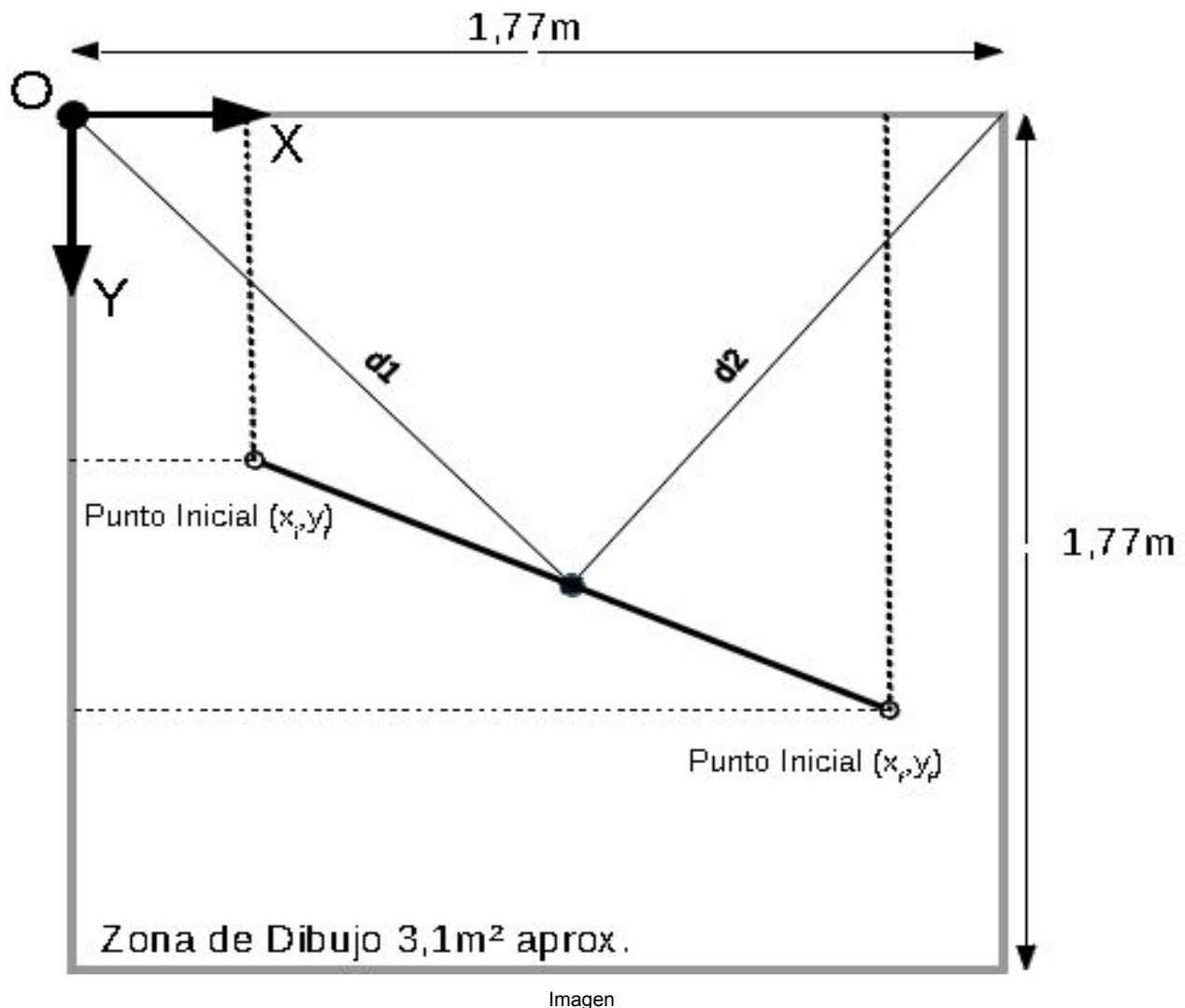
$$\begin{aligned}d_1 &= \sqrt{x^2 + y^2} \\d_2 &= \sqrt{(1768 - x)^2 + y^2} \\m_1 &= d_1 \frac{1 \text{ paso}}{1 \text{ revolución Motor}} \frac{360^\circ}{\pi \times \emptyset \text{ polea Motor}} \\m_2 &= d_2 \frac{1 \text{ paso}}{2 \text{ revolución Motor}} \frac{360^\circ}{\pi \times \emptyset \text{ polea Motor}}\end{aligned}$$

Donde para una resolución de rotación del motor de 1,8°, que en mi caso particular llevé a 0,9° utilizando pasos intermedios, y una polea de 5 mm de diámetro solidaria al eje del motor:

$$\frac{1 \text{ paso}}{0,9^\circ} \frac{360^\circ}{\pi \times 5 \text{ mm}} = 25,4648 \frac{\text{pasos}}{\text{mm}}$$

Esta constante representa la sección de círculo de la polea que el motor enrrollaría o desenrrollaría con la fracción de giro correspondiente a un paso. Vale la pena aclarar en este

momento que, como se puede ver en la expresión de esta constante, la resolución del dibujo (la capacidad de fraccionar una distancia en pedazos más pequeños) es inversamente proporcional a la resolución de movimiento del motor y al diámetro de la polea, por lo que con poleas más pequeñas o con una mayor resolución del motor (sea por características constructivas o por la implementación de técnicas como micropasos).



## 2.1. Control y comunicación

En cuanto al control se tuvieron las siguientes consideraciones:

1. Dado que un driver implementado con un microcontrolador no permite un procesamiento paralelo (y aunque lo proveyera sería muy difícil realizar de manera paralela el control del movimiento de cada motor), es necesario un driver particular para cada motor para poder realizar realmente desplazamientos coordinados.
2. Dado que los driver son independientes, la recepción de solicitudes del usuario (de un ordenador), el cálculo del paso de coordenadas, la comunicación de las mismas a los drivers y el coordinamiento de los movimientos debería también ser independiente.
3. También es necesario tener un control sobre el "dispositivo de impresión", aunque sea mínimo de encendido y apagado, dado que no en todos los desplazamientos se está "pintando".

Dadas las consideraciones se decidió una configuración Maestro-Esclavo, donde el Maestro se encarga de lo descrito en la consideración 2. Opté por una comunicación por UART con el usuario

y por un bus de comunicación I<sup>2</sup>C con los esclavos, con velocidad de transmisión de 100 KHz, ambos por tener un poco de experiencia previa con dichos protocolos. Además existe la posibilidad de que fuera más práctico tener el control del motor al lado del mismo, lo que implicaría PCBs separadas, y tengo entendido que la comunicación con protocolo SPI podría llegar a presentar limitaciones en cuanto a una comunicación de estas características, sin tener en cuenta que habría que tener 4 cables en vez de 2.

Decidí también asignar el control dicho en la consideración 3 al maestro por ser de momento sencilla, dado que no profundicé en el mismo, considero que si el dispositivo de escritura es más complicado de controlar que un “Encendido/Apagado” debería añadirse un esclavo particular para dicha tarea de control.

Los requisitos para el microcontrolador que oficia de maestro serían entonces las siguientes:

- Contar con módulo UART (Para la comunicación con el usuario)
- Contar con módulo MSSP (Para la comunicación con el bus I<sup>2</sup>C)

En cuanto a los requisitos para el esclavo:

- Contar con módulo MSSP (Para la comunicación con el bus I<sup>2</sup>C)
- Puertos de entrada/salida para el comando del motor y para la recepción de la señal del fin de carrera para la referenciación del motor.
- Contar con la posibilidad de Interrupciones por Timer para poder variar la velocidad del motor.

Se buscó en el catálogo online de microchip los microcontroladores que satisficieran dichos requerimientos, con la idea previa de utilizar un PIC16F876, por haber sido el utilizado en clases, pero el sitio de microchip recomendó el PIC16F886 porque el 876 se está descontinuando (<http://www.microchip.com/wwwproducts/en/PIC16F876> Imagen2).

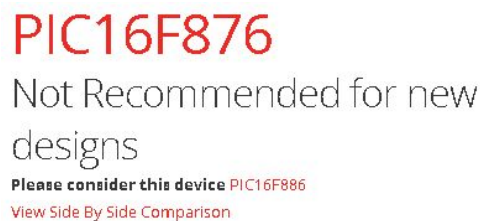


Imagen2

El PIC16F876 cumple tanto con los requerimientos para el rol de Master y para el rol de Esclavo. Para el control de cada motor se decidió implementar drivers independientes, cuya etapa de control se realiza con el microcontrolador antedicho y su etapa de potencia con un arreglo de transistores darlington ULN2803A.

## 2.2.Consideraciones constructivas

### 2.2.0.Dimensionamiento de la zona de trabajo

La limitación de la zona de dibujo radica en la cantidad de pasos de los que puede llevar cuenta el driver del motor. En este caso en particular, se utilizó la técnica de medio paso para tener una mejor resolución que con paso completo. Para contar los pasos del motor se decidió utilizar solamente una variable int16, la cual está limitada de 0 a 65535, es decir, puede contar hasta  $2^{16}$  pasos, entonces, para las características dichas en la sección de geometría:

$$65536 \text{ pasos} \frac{0,9^\circ}{1 \text{ paso}} \frac{\pi 5mm}{360^\circ} = 2573,6mm$$

Separados los motores esta distancia, el área para dibujar sería la intersección entre dos cuartos de circunferencias, cuyos centros están separados 2573,6 (Imagen3). Para contar con un margen de error y no trabajar al límite de las posibilidades, y para simplificar el cálculo, redondeé este

valor a 2,5 m. Pero me pareció que sería más representativo expresar esa área como un rectángulo, dentro de esta área el rectángulo más grande posible sería, siendo el área:

$$A = x_{\text{rectángulo}} \cdot y_{\text{rectángulo}}$$

Donde, llamando d a la distancia entre motores y al estar x acotada por las dos circunferencias

$$x_{\text{rectángulo}} = 2 \cdot \left(x - \frac{d}{2}\right)$$

La ecuación de y, es la de la circunferencia

$$y_{\text{rectángulo}} = \sqrt{2500^2 - x^2}$$

Entonces la ecuación del área, dependiente únicamente de x

$$A = 2 \cdot \left(x - \frac{d}{2}\right) \cdot \sqrt{2500^2 - x^2}$$

Para el caso d es 2500. Calculamos el área máxima

$$\frac{dA}{dx} = 2\sqrt{2500^2 - x^2} + \frac{2 \cdot \left(x - \frac{2500}{2}\right) \cdot (-2x)}{2\sqrt{2500^2 - x^2}}$$

$$0 = \sqrt{2500^2 - x^2} + \frac{x \cdot (-2x)}{2\sqrt{2500^2 - x^2}}$$

Despejando:

$$x^2 - 625x - 3125000 = 0$$

Ecuación cuya raíz positiva es:

$$x \approx 2107,676$$

Para la cual:

$$y \approx 1344.51 \quad y \quad A \approx 2.83m^2$$

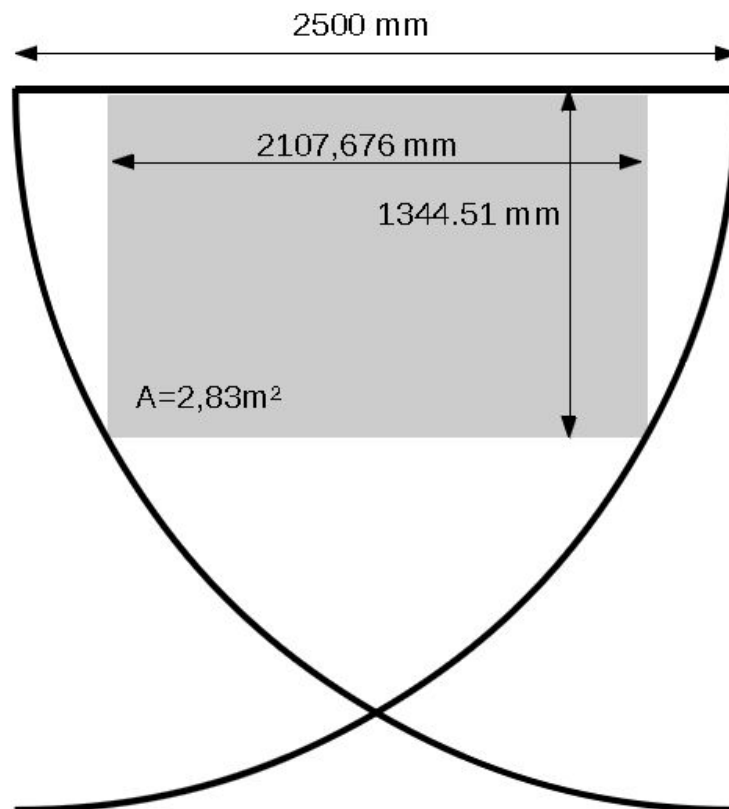


Imagen3

Pero en realidad ésta no es la configuración más óptima, si reducimos la distancia entre motores (Imagen4), las ecuación para el área sería la siguiente:

$$A = 2 \cdot \left(x - \frac{x}{2}\right) \cdot \sqrt{2500^2 - x^2}$$

$$A = x\sqrt{2500^2 - x^2}$$

Para el caso d es 2500. Calculamos el área máxima

$$\frac{dA}{dx} = \sqrt{2500^2 - x^2} + \frac{x \cdot (-2x)}{2\sqrt{2500^2 - x^2}}$$

$$0 = \sqrt{2500^2 - x^2} + \frac{x \cdot (-2x)}{2\sqrt{2500^2 - x^2}}$$

Despejando:

$$x = \frac{2500}{\sqrt{2}} \approx 1767.8$$

Para la cual:

$$y \approx 1767.8 \quad y \quad A \approx 3,1m^2$$

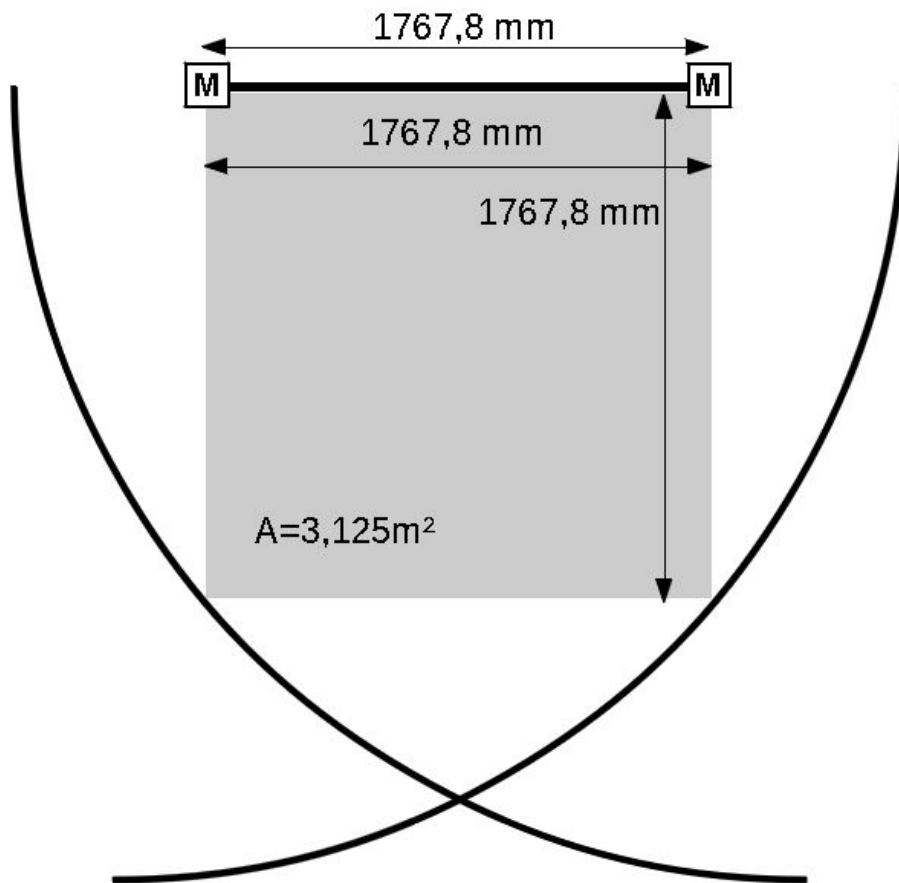


Imagen4

Ésta es el área máxima que podría tener para un control de medio paso limitado a un int16, para expandir ésta área se podría implementar una variable con límites mayores. Al principio de este informe se recomendaba para mejorar la resolución utilizar motores de mayor resolución o micro pasos, pero hay que tener en cuenta que esto podrías disminuir el área de dibujo

### 2.2.1.Cálculo de velocidades

Para coordinar el avance coordinado de los motores, la actualización de un paso al otro se implementó a través de interrupciones de sus timers. Antes de comenzar a dibujar una línea el maestro ordena a cada esclavo el valor al cual setear las interrupciones de su timer y así obtener las dos velocidades distintas para cada motor y llegar al punto final conjuntamente. Dado que estas son inversamente proporcionales a la cantidad de pasos a realizar, supongamos que al

Esclavo 1 tiene que realizar muchos pasos y el Esclavo 2 una cantidad menor, para aquel con la mayor cantidad de pasos a realizar (distancia más larga) se le ordena el tiempo de interrupción más corto posible (la mayor velocidad) y al otro esclavo :

$$CargaTimerEsclavo2 = 0xFFFF - \frac{pasosEsclavo2}{pasosEsclavo1} CargaTimerEsclavo1$$

### 2.2.2.Coordinación de los motores

Para intentar garantizar que ambos motores empiezan a desplazarse al mismo momento, ambos drivers tienen un puerto de entrada conectado a un fin de carrera que indica el contacto o el encendido del dispositivo de escritura, entonces al llegar al punto inicial ambos motores se detienen hasta obtener ésta señal y así lograr que ambos empiecen a desplazarse lo más simultáneamente posible.

## 3.Funcionamiento del sistema

### 3.0.Máquina de Estados

En las siguientes figuras se expresa la lógica del funcionamiento tanto del maestro como de los esclavos. Se describen gráficamente los diferentes estados que transcurren durante su funcionamiento así como las órdenes que los llevan a ellos. Vale la pena aclarar que las flechas de línea llena son aquellas que se realizan por una orden y aquellas de flechas con líneas a trazos representan el pasaje de un estado al otro por el normal funcionamiento del sistema y no por una orden externa. Por ejemplo, luego de realizar el proceso de búsqueda de referencia, el sistema sólo, sin necesidad de una orden externa, pasa al estado “Referenciado”.

En los diagramas también se muestran los mensajes, por UART o por I<sup>2</sup>C, que provocan el paso de un estado a otro.

En cada estado se muestra el valor de la variable de estado (un byte, ocho bits), en binario y en hexadecimal, en ésta variable cada bit representa una condición en particular del maestro o del esclavo. Para ambos se utilizó una variable cuyos bits tienen los siguientes significados:

Variable de Estado	-	-	D	L	M	H	R	P
bit	7...							...0

bit 0 ‘P’ :      0 Desenergizado  
                  1 Energizado

bit 1 ‘R’ :      0 No realizando Homing  
                  1 Realizando Homing

Dibujar

bit 2 ‘H’ :      0 Sin referenciar  
                  1 Referenciado

bit 3 ‘M’:      0 Quieto  
                  1 Movindose

bit 4 ‘L’:      0 No preparado para Dibujar  
                  1 Preparado para

bit 5 ‘D’:      0 No dibujando  
                  1 Dibujando

#### 3.0.0.Master

La máquina de estados correspondiente al maestro se ve en la Imagen5, el desarrollo de cada uno de los estados se desarrollará más adelante.



### 3.0.1.Slave

La máquina de estados correspondiente a funcionamiento de los esclavos se ve en la Imagen6, el desarrollo de cada uno de los estados se desarrollará más adelante.

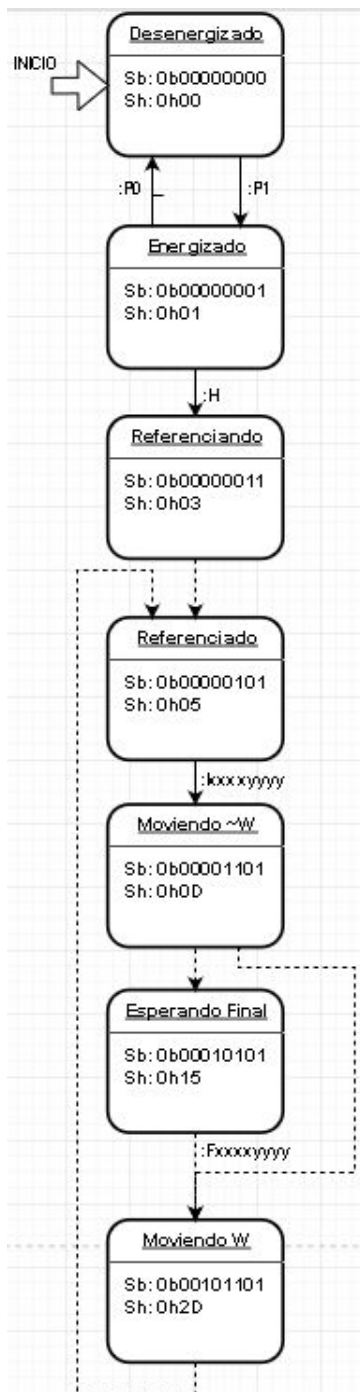


Imagen5

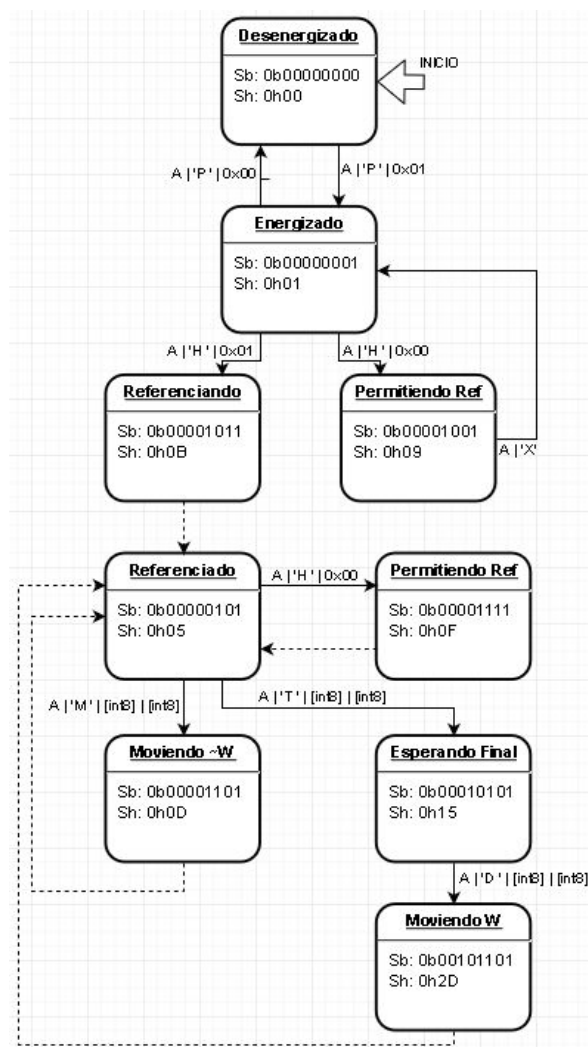


Imagen6

Vale la pena aclarar que en cuanto a las tramas de la comunicación I<sup>2</sup>C (Diagrama de Estados del Esclavo) A representa la dirección (Address), "|" el bit de acknowledge y aquello entre dos "|" es la información enviada en la transmisión (Ordenes, opciones o información).

## 3.1.Desarrollo de la comunicación y la secuencia de estados

### 3.1.0.Estado inicial - Desenergizado

Al inicio los motores se encuentran desenergizados. No hay tensión en sus bobinados, por lo tanto el eje está suelto o libre. La variable de estado del maestro vale 0x00 y la del esclavo también. El usuario debe solicitar Energizar mediante la orden “:P1”.

### 3.1.1.Energizado

Al recibir el maestro la orden “:P1”, ordena a los dos esclavos “A|’P’|0x01|0x00”, tras lo cual los esclavos también pasan al estado Energizado, donde ponen tensión en una de sus bobinas al azar. Al haber tensión en uno de los bobinados del motor el eje se fija a una posición. Las variables de estado del master, esclavo 1 y esclavo 2 valen 0x01. El usuario puede solicitar al master volver al estado Desenergizado con la orden “:P0” o referenciar el movimiento de los motores (Buscar referencia de inicio o hacer homing) con la orden :H.

### 3.1.2.Referenciando

En este estado el maestro se encarga de coordinar el referenciado de los dos motores.

#### 3.1.2.0.Referenciamiento del esclavo 1

El maestro informa al usuario que está por iniciar el referenciamiento del esclavo 1.

Primero ordena al esclavo 1 realizar su referenciamiento con la orden “A|’H’|0x01|0x00”, mientras que ordena al otro avanzar “A|’H’|0x00|0x00”, esto es para evitar generar una tensión entre las cuerdas que se encuentran unidas al dispositivo que realizará el dibujo. En dicho momento la variable de estado del master vale 0x03, la del esclavo 1 0x0B y la del esclavo 2 0x09.

Durante este estado el esclavo 1 pone su PIN C2 (pin de salida del esclavo conectado al pin de entrada C1 del maestro) en alto, lo cual indica al master que está ocupado desplazándose, y retrocede sin un objetivo definido. Cuando el esclavo uno detecta la señal del fin de carrera que indica el origen, deja de retroceder y avanza lentamente hasta dejar de percibir esta señal. Se detiene y determina en ese punto su origen (definiendo el valor de su posición actual a 0), pasa al estado Referenciado.

#### 3.1.2.1.Estado intermedio

Al pasar al estado Referenciado, variable de estado 0x05, pone en bajo su pin C2, señal que genera que el master ordene detenerse al esclavo 2, con la orden “A|’X’|0x00|0x00”. El esclavo 2 al recibir esta orden detiene su movimiento y vuelve al estado de Energizado. A continuación el master coordina el referenciamiento del esclavo 2.

#### 3.1.2.2.Referenciamiento del esclavo 2

El maestro informa al usuario que está por iniciar el referenciamiento del esclavo 2.

El master envía la orden al esclavo 1 de avanzar, para permitir al esclavo 2 buscar su referencia, con la orden “A|’H’|0x00|0x00”, con lo cual el esclavo 1 empieza a avanzar, sin perder su referencia, actualiza constantemente su objetivo 50 pasos adelante de su posición actual, variable de estado 0x0F. Luego el maestro envía al esclavo 2 la orden “A|’H’|0x01|0x00”, con lo que ejecuta los mismos pasos que el esclavo 1 en su proceso de referenciamiento. El pin C2 de salida del esclavo 2 esta unido al pin de entrada C3 del maestro.

### 3.1.2.3.Estado Final

Al el esclavo 2 pasar al estado Referenciado, variable de estado 0x05, pone en bajo su pin C2, señal que genera que el master ordene detenerse al esclavo 1, con la orden "A['X']0x00|0x00". El esclavo 1 al recibir esta orden detiene su movimiento y vuelve al estado de Referenciado, variable de estado 0x05. Finalmente el maestro pasa el estado Referenciado.

### 3.1.3.Referenciado

En este estado los el motor del esclavo se encuentra energizado y referenciado. El esclavo muestra al maestro que está desocupado poniendo en bajo su pin C2 en bajo. su variable de estado es 0x05. También deshabilitan las interrupciones de timer, lo que inhibe cualquier desplazamiento del motor.

Una vez referenciados los desplazamientos de los dos motores, se puede empezar a dibujar líneas.

El maestro informa al usuario que está listo para recibir las coordenadas de la línea a dibujar. Su variable de estado es 0x05.

### 3.1.4.Movimiento sin dibujar

Al recibir la coordenada inicial de la línea a dibujar con el formato ":lxxxxyyyy", donde "xxxx" e "yyyy" son las coordenadas x e y del punto inicial, el maestro realiza la conversión de coordenadas como se explicó en la sección 2.0.Geometría y comunica a cada esclavo sus coordenadas correspondientes con las órdenes "A['M'] [int8-H] | [int8-B]". Dado que el valor de las coordenadas son int16, para enviarlas, el maestro las separa en el byte de mayor peso "[int8-H]" y el de menor peso "[int8-B]", las envía separadas y al recibirlas, cada esclavo la reconstruye.

Al recibir la orden los esclavos actualizan su posición objetivo y habilitan las interrupciones de timer (lo que les permite actualizar los valores del puerto de salida que controla la tensión en los bobinados) con las interrupciones lo más cortas posibles (desplazamiento rápido). También indican al maestro que se están desplazando con respectivos pines C2.

Durante este estado las variables de estado del maestro y los esclavos valen 0x0D.

Al llegar a la posición objetivo los esclavos vuelven al estado Referenciado (variable de estado 0x05), donde indican al maestro que están desocupados.

Al recibir esta señal de ambos esclavos (pines de entrada C1 y C2 en bajo) el maestro entiende que ambos esclavos están en posición para empezar a dibujar. El maestro avanza al estado Esperando punto final.

### 3.1.5.Esperando punto final

El maestro informa al usuario que el dispositivo para dibujar se ha posicionado en el punto inicial de la línea, y necesita el punto final para continuar. La variable de estado del maestro vale 0x15.

### 3.1.6.Dibujando

#### 3.1.6.0.Preparándose para dibujar

Al recibir del usuario la orden ":Fxxxxyyyy", donde "xxxx" e "yyyy" son las coordenadas x e y del punto final, el maestro pasa al estado de dibujar. Su variable de estado es 0x2D. Mientras los esclavos siguen sin moverse en la posición inicial en el estado de Referenciado.

Primero el maestro calcula las coordenadas en pasos del punto final como se explicó en la sección 2.0.Geometría. Luego calcula las velocidades a las que deberían avanzar los pasos de

cada esclavo, como se explicó en la sección 2.2.1.Cálculo de velocidades. Comunica las velocidades correspondientes a cada esclavo con la orden “A|‘T’| [int8-H] | [int8-B]”. Dado que ambos esclavos utilizan el Timer1 de 16 bits, para enviarles el valor al cual deben configurar las interrupciones del mismo, el maestro las separa en el byte de mayor peso “[int8-H]” y el de menor peso “[int8-B]”, las envía separadas y al recibirlas, cada esclavo la reconstruye. Al recibir esta orden, los esclavos pasan del estado Referenciado (variable de estado 0x05) al estado Esperando las coordenadas del punto final (variable de estado 0x15).

Luego el maestro envía las coordenadas correspondiente a cada esclavo del punto final con la orden “A|‘D’| [int8-H] | [int8-B]”. Al recibir esta orden los esclavos pasan al estado Dibujando (variable de estado 0x2D).

#### 3.1.6.1.Dibujando

En el estado Dibujando, los esclavos deben esperar la señal de encendido del dispositivo que ejecuta el dibujo, señal digital que ingresa por el puerto de entrada C1. Dicho dispositivo es comandado por el maestro por su puerto de salida C0.

Entonces, luego de enviar las coordenadas del punto final de la línea a dibujar, activa el dispositivo de dibujo (puerto C0 en alto). Los esclavos reciben alto en su pin C1, habilitan las interrupciones de su timer, lo que habilita el avance del motor, y ponen sus respectivos pines C2 de salida en alto, indicando al maestro su desplazamiento.

#### 3.1.6.2.Fin del dibujo

Al alcanzar la posición objetivo, ambos ponen sus pines C2 de salida en bajo y pasan al estado de Referenciado (variable de estado 0x05). El maestro percibe que ambos esclavos han terminado porque sus pines de entrada C1 y C2 se ponen en bajo, entonces apaga el dispositivo de escritura (pin de salida C1 en bajo). A continuación pasa al estado de Referenciado (variable de estado 0x05) e informa al usuario que se terminó de dibujar la línea ordenada y que está a la espera de órdenes.

#### 3.1.7.Solicitud de estado

En cualquier momento, el usuario puede solicitar al Master su estado o el de los esclavos. La orden “:S0” para solicitar el estado del maestro, “:S1” para el estado del esclavo 1 y “:S2” para el estado del esclavo 2. Además tanto maestro como esclavo muestran sus estados a través el puerto A.

## 4.Elección y Cotización de Elementos

### 4.0.Motores

Para el diseño y dimensionamiento del sistema se tomó como referencia las características del motor paso a paso bipolar PKP213U05A, no solamente porque satisface los requerimientos, sino también porque pudo conseguirse basta información de otros aspectos del motor. Como ya se mencionó con anterioridad, tiene una resolución de 1,8°. Precio: USD 55,25. Otras características pueden ser vistas en:

<http://catalog.orientalmotor.com/item/stepping-motors--1068/pk-series-stepping-motors/pkp213u05a?&plpver=1068&origin=keyword&by=prod&filter=0#> .

## 4.1. Microcontroladores

Como se explicó en la sección 2.1. Control y comunicación, se seleccionó el PIC16F886 de Microchip, tanto para la función de maestro como de esclavo. Su costo en la actualidad oscila entre los ARS100 y ARS130. Más información en:

<http://www.microchip.com/wwwproducts/en/PIC16F886>

## 4.2. Piezas 3D

Para completar el funcionamiento del sistema se debió realizar algunas piezas a medida.

### 4.2.0. Mordaza para dispositivo de escritura

Para fines de prueba del sistema de control de posición se considera conveniente probar con un marcador para pizarra sobre una superficie en la que pueda borrarse los trazos. Para ello se diseñó en SolidWorks el sistema de “mordazas” que se muestra en la Imagen7.

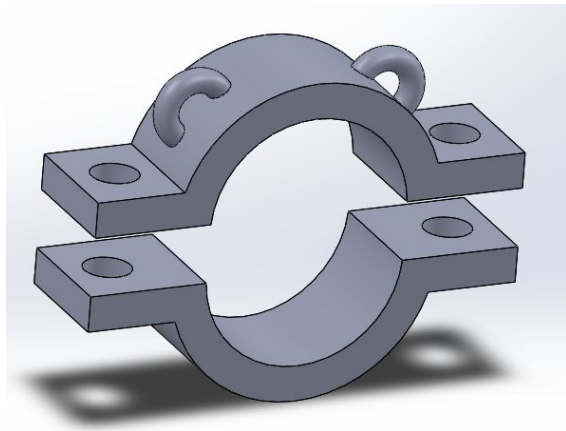


Imagen7

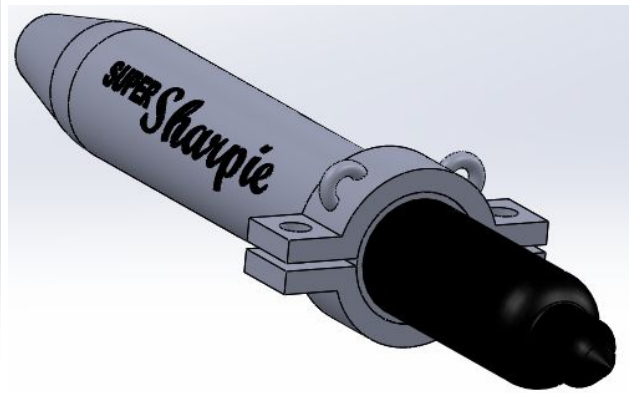


Imagen8

Las “orejas” están posicionadas a 45° respecto a la vertical. Como se muestra en la Imagen8, serviría para aprisionar un marcador y unirlo mediante las orejas a las cuerdas que se enrollan en los ejes de los motores.

El costo aproximado de la impresión 3D es de ARS20, ARS10 aproximadamente cada mitad de la mordaza.

### 4.2.1. Poleas

Tanto para ser colocadas en los ejes de los motores y sujetar las cuerdas (para enrollar y desenrollarlas) como para ser colocados en los vértices superiores del área de dibujo para cambiar la dirección de las cuerdas, se hicieron, también en SolidWorks, unas poleas de un diámetro interior de 5mm (Imagen9). Como se puede apreciar en la vista en corte de la Imagen 0, posee una perforación a través de la polea, para permitir fijar la cuerda y perforaciones en la zona que abrazaría al eje, para permitir fijarlo al mismo.

El costo aproximado de la impresión 3D es de ARS10 cada polea. Se necesitarían 4 de ellas.

### 4.2.2.Ensamble

Con los modelos desarrollados más el modelo de un soporte y modelos importados de motores paso a paso similares al descrito se armó un ensamble para poder tomar las dimensiones del área de dibujo respecto a los componentes. Dicho ensamble se muestra en las Figuras 11, 12, 13 y 14.

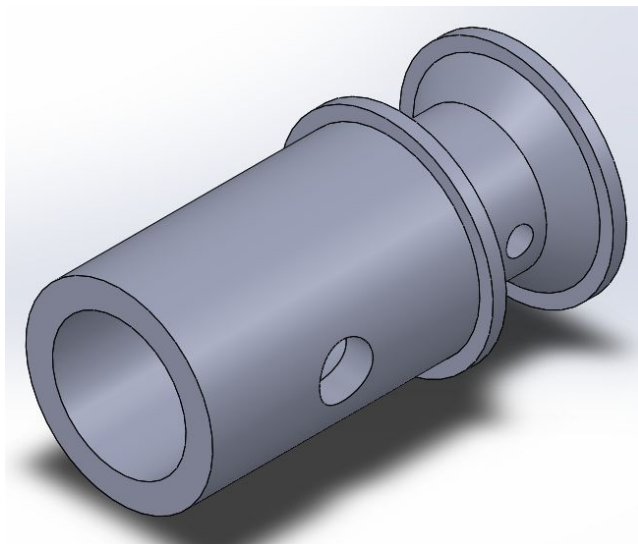


Imagen9

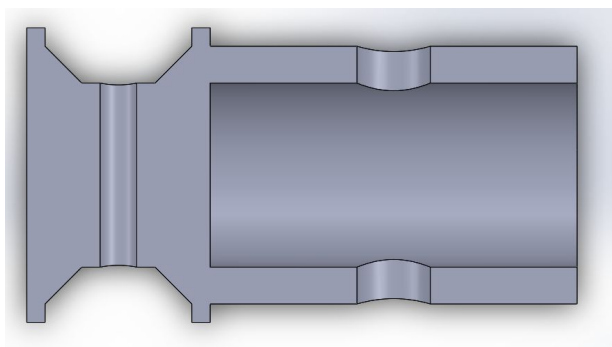


Imagen 0

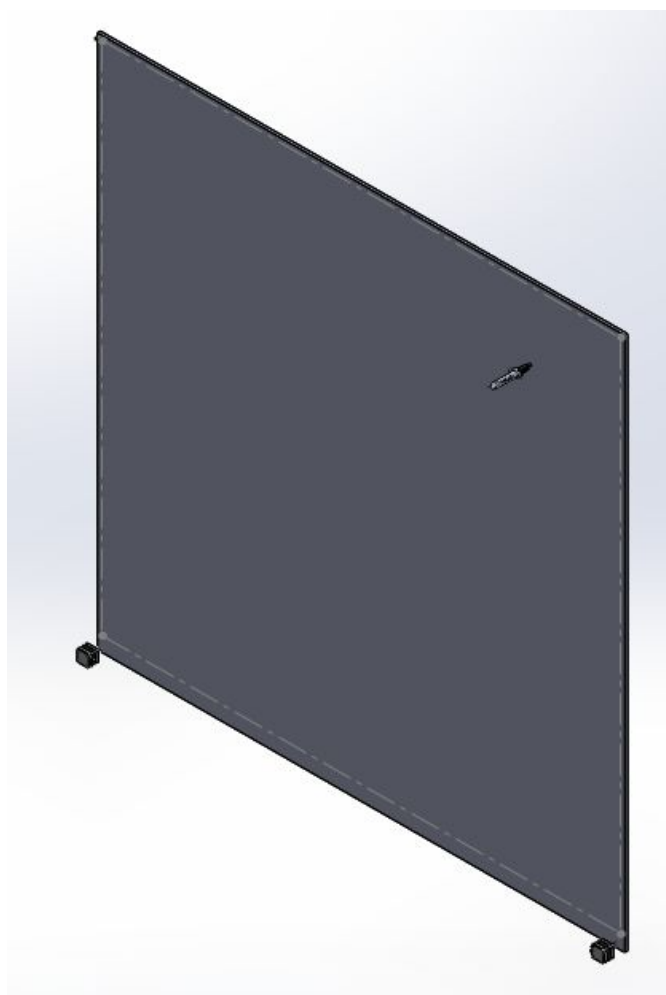


Imagen 1

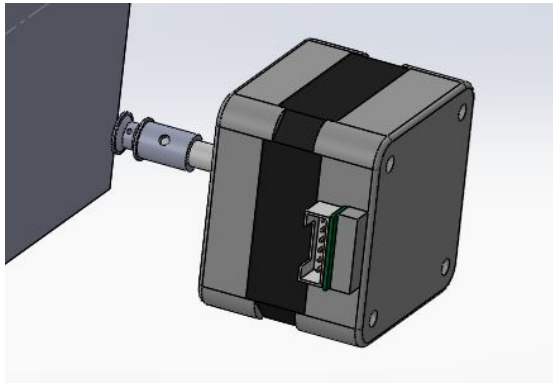


Imagen 2

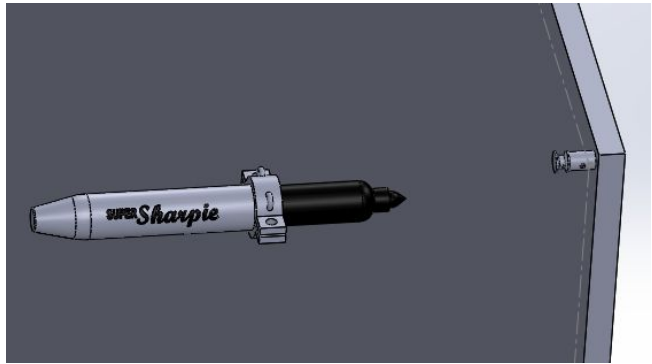


Imagen 3

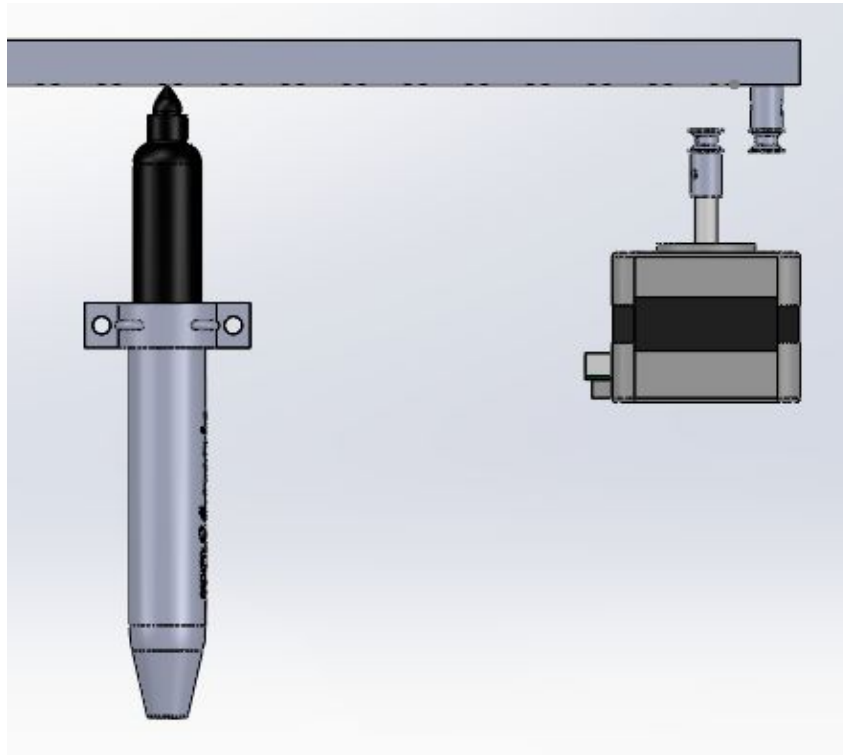


Imagen 4

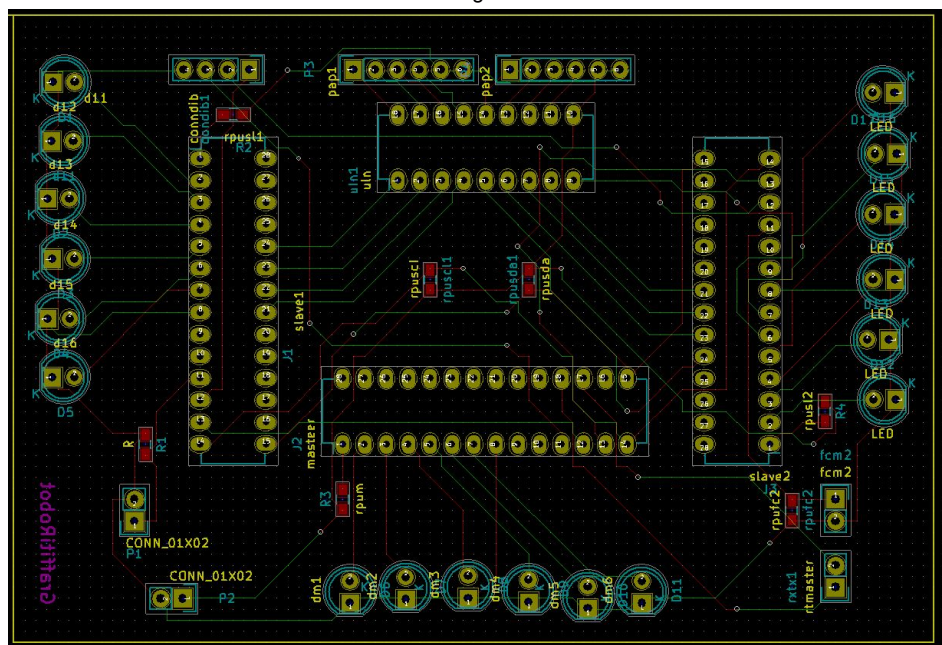


Imagen 5

### 4.3.PCB

Con la herramienta KiCad se realizó el diseño de una PCB de doble cara (Imagen 15) que agrupa la fase de control (maestro y dos esclavos) y la fase de potencia (ULN2803A), mediría aproximadamente 95x69mm. Tiene dispuesto los conectores para la comunicación serie, así como para los fines de carrera y los motores.

Además se incorporaron leds para poder mostrar visualmente la variable de estado de cada esclavo y del maestro.

Se generaron todos los archivos necesarios para su construcción sea en casa sea a pedido. El costo aproximado es de USD17 (<https://oshpark.com/>).

## 5.Repositorio GitHub

Todos los documentos relacionados al presente trabajo se encuentran en un repositorio en GitHub. En él (<https://github.com/nahueJ/graffitiRobot>) se podrán encontrar los siguientes archivos, ordenados de la siguiente manera que se muestra en la Imagen 6.

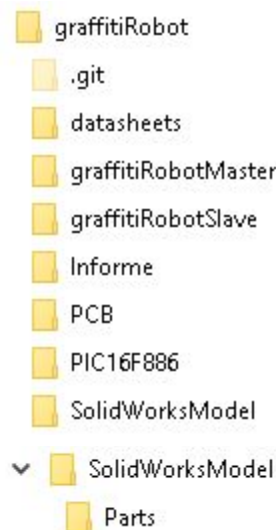


Imagen 6

- En la carpeta principal se encuentra el proyecto de Proteus, que contiene el esquemático utilizado para probar y depurar el funcionamiento del sistema.
- Dentro de la carpeta **datasheets** se encuentran la documentación e información recolectada concerniente a los componentes electrónicos y motores.
- Dentro de las carpeta **graffitiRobotMaster** y **graffitiRobotSlave** se encuentran los códigos fuente (.c) y archivos de encabezado (.h), así como los archivos productos de la compilación (.cob, .hex, etc.) de los programas escritos para el maestro y los esclavos desarrollados con el editor de texto Sublime3 y el Compilador PCWHD.
- Dentro de la carpeta **Informe** se encuentra el presente texto.
- Dentro de la carpeta **PCB** se encuentran todos los documentos relacionados al diseño realizado con la herramienta KiCad del circuito impreso, tanto el proyecto de KiCad como los códigos Gerber de las diferentes capas.
- Dentro de la carpeta **SolidWorksModel** se encuentran los documentos relacionados al modelo mostrado en las Figuras 11, 12, 13 y 14.
  - Dentro de la subcarpeta **Parts** se encuentran los documentos de las piezas que forman parte del ensamble citado en el ítem anterior.



## 6.Trabajo a futuro

El presente trabajo es el producto de una propuesta de proyecto para complementar e implementar control de posición, comunicación ordenador - microcontrolador, así como microcontrolador - microcontrolador y aplicación de transistores bajo régimen de conmutación. Las cuales fueron temáticas abordadas durante el cursado.

Dejando de lado esto, para un mejor funcionamiento del sistema (si es que existe una demanda del mismo y en caso de haber deberían evaluarse los pormenores de dicha demanda), es necesario que el control de posición y velocidad, que en este modelo se realiza a lazo abierto, se realice a lazo cerrado.

Durante este trabajo no se profundizó en la selección o diseño de los fines de carrera que se adaptan mejor al sistema, que podrían ser mecánicos o magnéticos, sobre los cuales se debería evaluar la precisión, rendimiento y relación costo - beneficio de los mismos.

Debería probarse, en función del dispositivo que ejecute el dibujo, cuál es la resolución que se necesita realmente. Dado que no será igual la resolución que se requiere para realizar el dibujo con un lápiz o un marcador, que con un aerosol por ejemplo, que posee un trazo mucho más grande.

Tampoco se profundizó en el soporte del elemento para efectuar el dibujo. Si bien se realizó un rápido y sencillo para sostener un lápiz o un marcador, en caso de ser requerido un sistema a aerosol o con un aporte continuo de tinta, como un aerógrafo, el soporte se tornaría no solo más voluminoso, si no también más complejo. En cuyo caso había un trabajo de selección y diseño de soporte y actuador para el dispositivo en cuestión, y por otro lado se debería evaluar si el torque de los motores es suficiente para posicionar dicho dispositivo.

Finalmente, si bien se intentó que la comunicación fuera lo más intuitiva posible, indicando el inicio y fin de los procesos, solicitando el ingreso de información, indicando en caso de error la sintaxis correcta para ingresar los datos. La comunicación entre el usuario y el maestro no es lo mejor que puede llegar a hacerse al respecto. Considero que si este sistema se pensara para un uso de personas sin formación en protocolos de comunicación, se debería realizar un desarrollo de software que fuera capaz de tomar los datos de un programa de dibujo o tal vez capaz de procesar un formato de dibujo vectorizado, y dicho software sería el encargado de llevar a cabo la comunicación con el microcontrolador y no el usuario.