

Proyecto de Informática - Plotter CNC

Autores: Miguel Ángel Asensio Vaquero
Rodrigo Batal Fernández

Índice: 1-Proyecto
2-Componentes principales
3-Funciones
4-Código y pinout del Arduino

1-Proyecto

En un principio, el proyecto consistía en fabricar y hacer funcionar un modelo semejante a una grúa de feria. Finalmente, optamos por crear una especie de plotter con un funcionamiento similar al de una impresora (dibuja en dos ejes x e y, subiendo y bajando un lápiz), debido a que resultó más sencillo de fabricar.

El plotter está construido a partir de un par de lectores de DVD (puesto a que nos dejaban el trabajo hecho de crear una estructura que nos permitiese un desplazamiento lineal), y una estructura de impresión en 3D para poder subir y bajar el boli.

En lo que se centró el proyecto fue en el control de motores de tipo stepper. A diferencia de los motores DC convencionales funcionan imantando y desimantando las diferentes bobinas que constituyen el motor, logrando así que el eje del motor siga orientaciones muy concretas. Esto permite a su vez un gran control en la posición del eje en todo momento, lo que facilita la deducción de la posición del boli dentro de un marco de referencia.

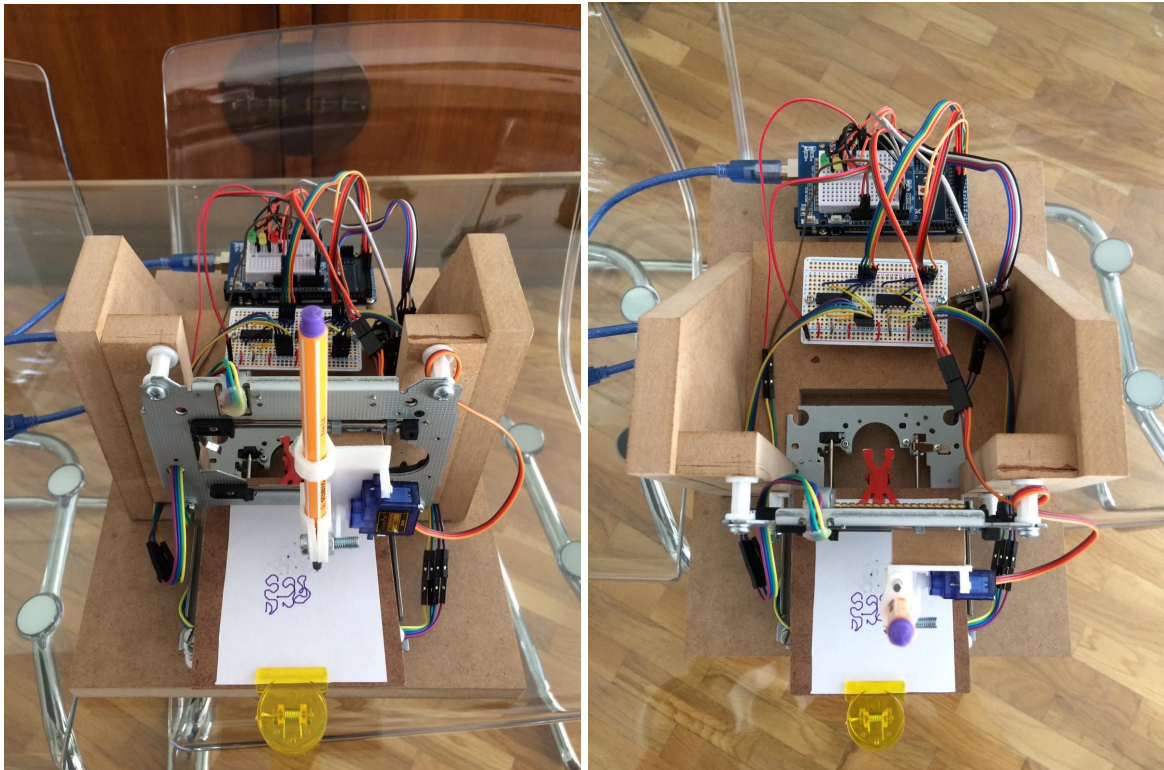
Cabe destacar que la comunicación con los dispositivos de Arduino difiere en cierta medida de los contenidos tratados en clase, por lo que ha sido necesario un aprendizaje externo. También tratamos de enlazar Arduino con Visual Studio para así trabajar exclusivamente con los conocimientos obtenidos durante el curso pero, debido a las complicaciones del proceso, decidimos trabajar en Arduino directamente. Por este motivo, pese a que teníamos pensado trabajar con ficheros en el proyecto, finalmente, no fue realizable (a diferencia de Visual Studio, Arduino no trabaja con ficheros).

Otro problema destacable es el hecho de que el buffer en Arduino debía ser limpiado con frecuencia llegando a leer e imprimir “basura” como solución definitiva cada vez que quisiéramos leer por teclado y escribir por pantalla.

Los problemas surgidos con el intento de vincular el Arduino con Visual Studio de deben a la dificultad que existe con la comunicación entre dispositivos a través del Serial port; esto es porque el Arduino se comunica en formato byte, limitando así en gran medida el volumen de la información, sobretodo la recibida.

Fue posible enviar y recibir datos en formato char, pero con gran dificultad debido a los errores de buffer que se acumulan poco a poco con el tiempo. Por ello, al final, la intención de que fuese posible leer un fichero con una colección de vectores y que el ordenador se lo transmitiese al Arduino para lograr dibujos más complejos quedó en un intento fracasado.

Proyecto de Informática - Plotter CNC



2-Componentes principales

- Arduino Mega 2560 (para controlar el proyecto).
- Dos motores stepper (extraídos de lectores de CDs).
- Motor servo (para controlar la subida y bajada del lápiz).
- Finales de carrera (para limitar el movimiento de los motores en el eje x e y).
- Madera (estructura).
- Cableado y componentes electrónicos menores.

3-Funciones

Las funciones se dividen en dos grupos: las de control de los motores (3.1), y las de acciones preparadas (3.2). Las de control de los motores se centran en el imantado-desimantado de las bobinas para controlar el movimiento, así como de la “percepción” de la posición en función de los pasos dados desde el punto inicial del recorrido. Mientras, las de acciones preparadas son aquellas que dan uso a las de control de los motores para lograr generar el dibujo deseado.

3.1-Funciones de control de motores

-Función “mueve”

Esta función se dedica a controlar el paso de cada motor (la secuencia de encendido-apagado del bobinado). Sabemos que una vuelta completa del motor está

Proyecto de Informática - Plotter CNC

compuesta de 8 pasos a partir de combinaciones de encendidos y apagados de los cuatro inputs del motor; de esa forma, logramos que el motor avance siguiendo la secuencia en un sentido, y que retroceda siguiendo la secuencia en el opuesto.

Al iniciar la función, esta recibe como argumentos el desplazamiento deseado, y el puntero respectivo del eje al que se quiere aplicar. Al evaluar la estructura que contiene el puntero del eje, se puede extraer como información de qué eje se trata (x o y), cuál es su coordenada actual, y en qué paso está el motor en ese mismo momento.

Las unidades longitud aplicadas a las coordenadas que el Arduino te pide han sido hechas en función de la distancia recorrida por un paso, logrando así una proporción 1:1, que facilita la ejecución del programa.

En el caso de que el soporte del boli se encuentre en el extremo del recorrido, y el Arduino es consciente de ello (vía los finales de carrera, o el contador de coordenadas), por mucho que se le imponga avanzar, no lo va a hacer, impidiendo que el dibujo que se esté haciendo se estropee, y que la estructura reciba daños al ser forzada.

Conforme al desplazamiento del soporte del boli por el eje, la función va sobrescribiendo tanto la posición del eje como el paso del motor, teniendo así una percepción permanente (si no falla el desplazamiento del eje) de la posición de la estructura. Cabe destacar que el desplazamiento real no tiene por qué corresponderse con el introducido, si no es posible efectuarlo (si se llega al límite del recorrido).

-Función "ir_a"

La función "ir_a" llama a la función "mueve" para controlar el movimiento de los motores, dirigiéndolos a unas coordenadas específicas.

Recibe como argumentos las coordenadas deseadas, y los punteros de las estructuras de control de los ejes. Extrayendo la posición actual en ambos ejes a través de los punteros a las estructuras, junto con las coordenadas deseadas introducidas en el argumento, calcula el desplazamiento necesario en los ejes para pasar de la posición inicial a la especificada.

Mediante la función "mueve", realiza el desplazamiento íntegro, primero respecto al eje x, y luego respecto al eje y.

-Función "ir_recto_a"

La función "ir_recto_a" llama a la función "ir_a", para unir dos puntos (el actual y el marcado) mediante una línea recta. Nos permite ir en una línea "recta" siguiendo la ecuación de una recta a partir de dos puntos de tal manera que los motores realicen el movimiento horizontal y vertical de "ir_a" del tamaño mínimo posible, alcanzando una gran precisión, dando la impresión de ser una recta uniforme.

La función en un principio evalúa cuál es el mayor desplazamiento (si respecto al eje x o al y), para hacer la recta en función del eje con el mayor desplazamiento, pues al consistir en un conjunto de "ir_a" con desplazamientos unidad en función de uno de los dos ejes, se consigue así un mayor refinado; además, así se logran evitar las rectas de pendiente infinito, que confunden al sistema.

La recta queda así "escalonada", pero de la forma más mínima posible. Dado que estamos programando en C, al no poder usar programación sobre objetos, no hay forma de

Proyecto de Informática - Plotter CNC

controlar los dos motores de forma simultánea, con lo que el escalonado y la ecuación de la recta resultaron la opción más precisa viable.

-Función “ir_a_referencia”

Con esta función conseguimos que (mediante los finales de carrera) los motores se desplacen hasta una posición de referencia (0,0) para así poder empezar a dibujar desde cero, independientemente de cómo se encuentren los motores en un principio.

Para ello, al inicio de la función, marcamos artificialmente la posición de la estructura como (1000,1000), muy por encima del límite (para que pueda retroceder libremente sin que salten los bloqueos de protección), y tras ello va retrocediendo poco a poco hasta que toca los finales de carrera del (0,0), marcando esa posición como el origen.

Si tras avanzar un total de (600,600) unidades no logra activar los finales de carrera (la cuadrícula de movimiento del boli es de 450*450), la secuencia de error de sistema salta, frenando toda actividad, y tras un rato reseteando el sistema.

Al inicio del programa el paso de los motores también es desconocido, con lo que se marca artificialmente al inicio de la secuencia, y conforme a se va retrocediendo en la misma, en algún punto el eje se alineará correctamente con la secuencia del bobinado; a partir de entonces siempre que algún paso esté marcado, el eje estará bien alineado.

Por errores en el montaje de la maqueta, actualmente se ha modificado la función para buscar el (0,450), pues solo dos finales de carrera de los cuatro programados han sido instalados, y de ellos han sido puestos el de inicio del eje x y el del final del eje y.

3.2-Funciones de acciones preparadas

-Función “segmentos”

El usuario debe indicar el número de puntos que desea unir (por asignación dinámica de memoria) e introducir las coordenadas de esos puntos para unirlos mediante segmentos. Si los puntos no estuviesen dentro del rango indicado, volverían a ser solicitados. Una vez introducidos correctamente, el plotter procederá a marcar los puntos uno a uno haciendo uso de la función “ir_a” y, posteriormente, los unirá haciendo uso de la función “ir_recto_a”. Al dar por terminada la función se devuelve el lápiz a su posición límite.

-Función “lagrange”

El usuario debe indicar el número de nodos que desea estudiar (por asignación dinámica de memoria) e introducir las coordenadas de esos nodos para dibujar su respectivo polinomio de Lagrange. Si los nodos no estuviesen dentro del rango indicado, volverían a ser solicitados.

Una vez introducidos correctamente, a partir de una serie de bucles se genera una nube de puntos alojada en un vector del tamaño de la extensión del rango de movimiento del eje x, que se corresponde al valor del polinomio en cada coordenada respecto a dicho eje (si las

Proyecto de Informática - Plotter CNC

coordenadas de algún punto se encuentran rebasando el límite del entorno de dibujo, se le asocia el límite más cercano).

Tras esto, el plotter procederá a marcar los nodos uno a uno haciendo uso de la función "ir_a" y, posteriormente, realizará el dibujo del polinomio recorriendo la nube de puntos del menor al mayor respecto al eje x con un "ir_a". Al dar por terminada la función se devuelve el lápiz a su posición límite.

-Función "joystick"

Esta función permite al usuario controlar el movimiento del plotter mediante un joystick externo. La función mapea los valores recibidos del joystick analógicamente para controlar los movimientos de los motores en los ejes **x** e **y**. Según la inclinación del joystick los motores permanecerán en reposo, se desplazarán con velocidad 1 o con velocidad 2 (a mayor inclinación mayor velocidad). El botón del joystick controla la subida y bajada del servo y, por lo tanto, del lápiz. De esta manera, se podrá dibujar (virtualmente) cualquier forma que se desee, siempre que se tenga el manejo suficiente del joystick. Al introducir cualquier valor o caracter termina la función devolviendo el lápiz a su posición límite.

3.3-Otras funciones

-Función "sys"

Controla tres LEDs que indican el status del sistema: Verde (OK), Amarillo (en proceso), Rojo (ERROR). Si se da el status de error se dispara el reseteo del sistema.

-Función "cuadrícula"

Mediante el uso de la función "mueve" e "ir_a" en dos bucles el plotter dibuja una cuadrícula para que los polinomios de Lagrange puedan apreciarse mejor al ser más visual. Está compuesta por un eje **x** y un eje **y** divididos de 50 en 50 unidades (de 0 a 450).

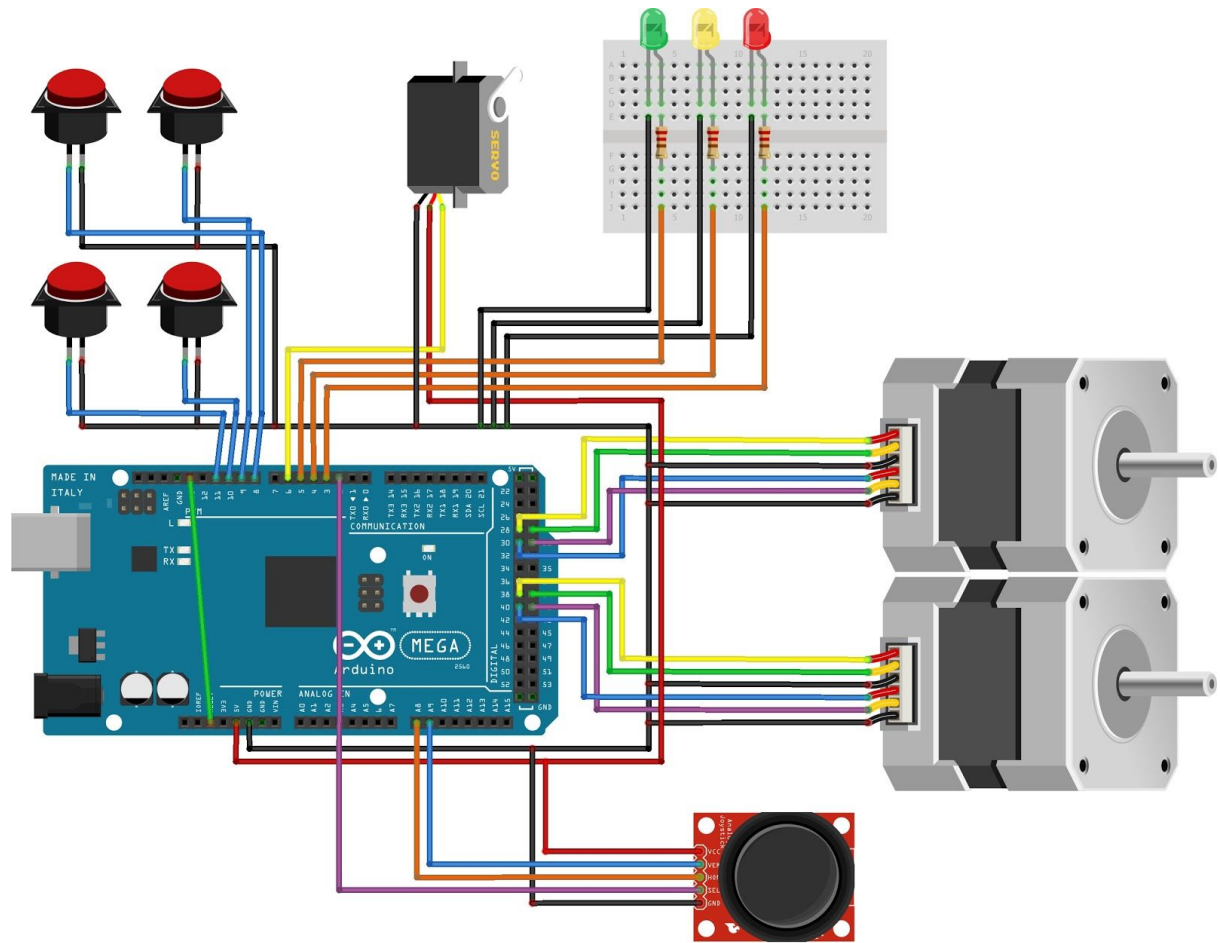
4-Código y pinout del Arduino

Este es el código incorporado en el Arduino Mega 2560, con comentarios para una mejor interpretación.

El código se encuentra en la carpeta adjunta llamada CNC_Plotter por motivos de extensión y formato.

Este es el pinout del Arduino, simplificado para motores stepper unipolares (para una mejor comprensión); para motores bipolares es necesario incluir un puente-h por motor para poder así invertir el voltaje en las bobinas. A efectos prácticos, el funcionamiento es literalmente el mismo.

Proyecto de Informática - Plotter CNC



fritzing