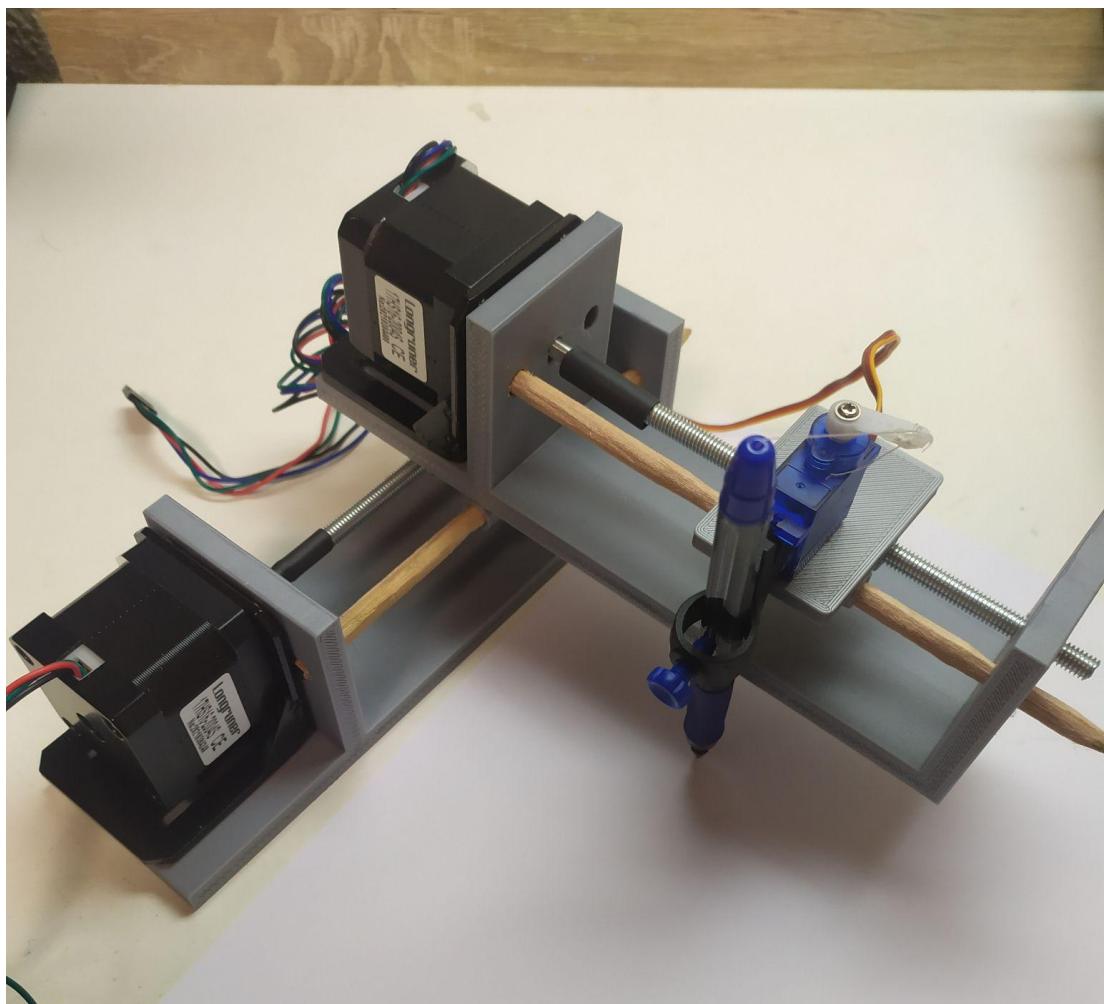


# PROYECTO ARDUINO: IMPRESORA

*Trabajo de impresora con Arduino*



**Sergio Guillén y Eduardo Guerrero**

17/03/2022

2.<sup>º</sup> H, Informática

# ÍNDICE

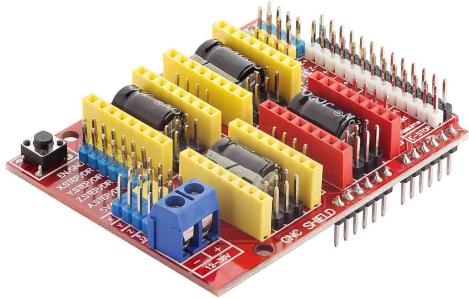
<b>INTRODUCCIÓN</b>	<b>1</b>
<b>MATERIALES</b>	<b>1</b>
<b>PROCEDIMIENTO</b>	<b>3</b>
Convertir .png a .ng	3
Conectar GRBL a arduino	4
Conectar CNC shield y motores	5
GRBL Controller	5
Modificar comandos	6
Conectar el bluetooth	7
<b>MECANISMO</b>	<b>7</b>
<b>PROBLEMAS</b>	<b>8</b>
<b>COSAS A MEJORAR</b>	<b>9</b>
<b>CONCLUSIÓN</b>	<b>9</b>

## INTRODUCCIÓN

Nuestro proyecto de arduino trata de una impresora, hemos elegido este proyecto porque nos parecía original, vistoso y nos podría suponer un reto con el que aprender de arduino.

## MATERIALES

- Arduino
- CNC Shield

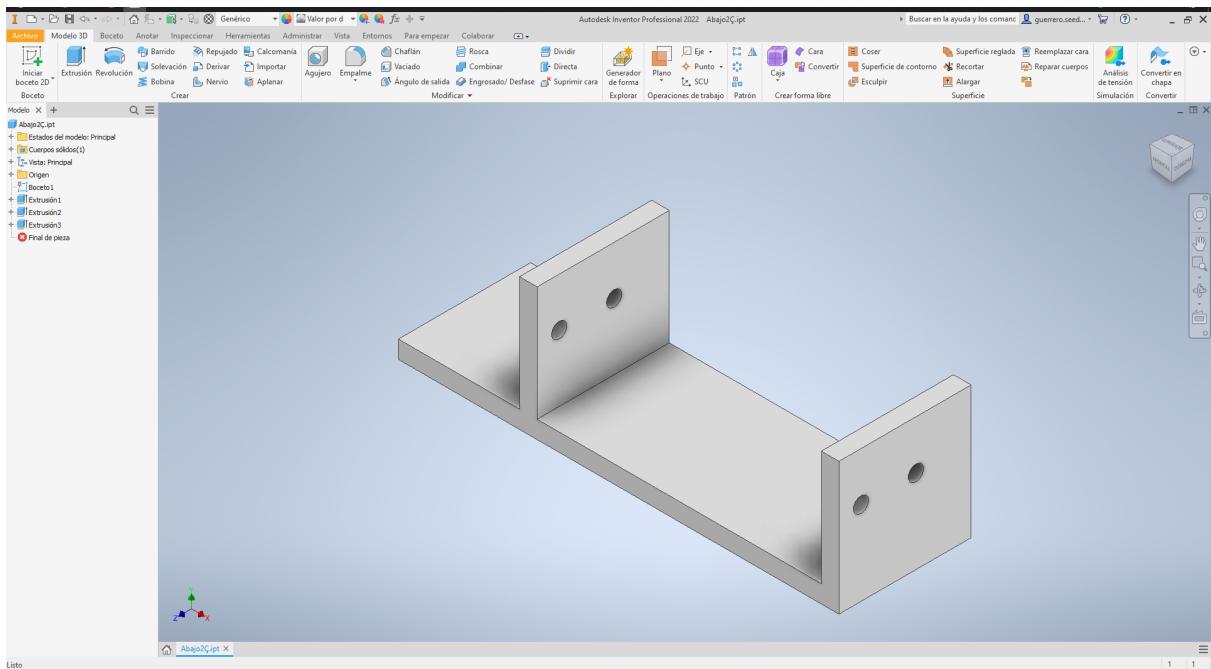


- Drivers A4988 (x3)
- Mini jumpers (x6)
- Motor nema (x2)



- Servo motor

- Piezas con impresora 3D

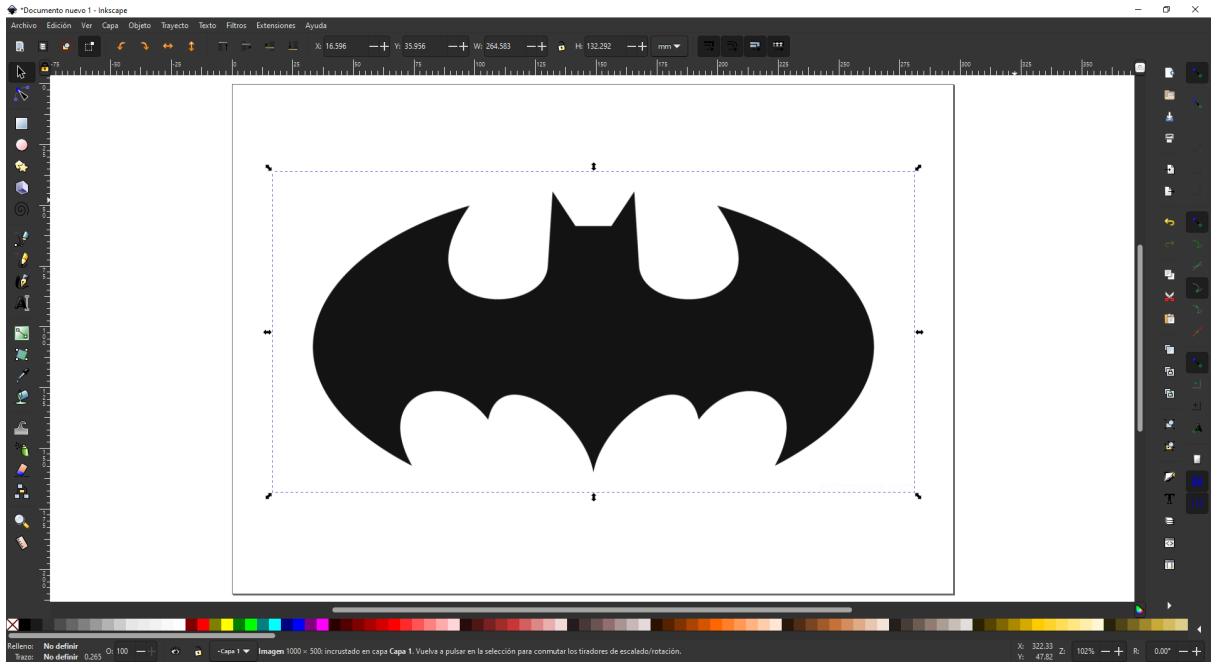


- Palo con rosca
- Tuerca 6 mm (x4)
- Lápiz (x2)
- Punta de compás
- Bolígrafo con muelle

## PROCEDIMIENTO

### 1. Convertir .png a .ng

Primero, convertir la imagen .png a una **imagen vectorizada** con formato .ng o .ngc, esto lo hacemos para que el programa pueda leer las líneas del contorno. En nuestro caso hemos elegido el programa de **Inkscape** para vectorizar nuestra imagen.



## 2. Conectar GRBL a arduino

Conectar solo el arduino al ordenador (si conectamos también la CNC shield podría sobrecalentar algún componente y estropearlo) e iniciar el grbl. Esto se hace descargando una **librería llamada grbl** y el código ya te sale en el programa de arduino en  
Archivo>Ejemplos>GRBL-Arduino-Library>GRBLtoArduino

Este es el código que te importa:

```

/*
Thanks for supporting Open-Hard/Soft-ware and thanks
for all of the contributors to this project.

For extra info on GRBL please have a look at my blog :
http://blog.protoneer.co.nz/tag/grbl/

Grbl is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

Grbl is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.
http://www.gnu.org/licenses/
*/

/*
Supported hardware:
  Arduino Duemilanove
  Arduino Uno
  Arduino Mega 2560 (Limited Testing)

*/
#include <grblmain.h>

void setup(){
  startGrbl();
}

void loop(){}

```

### 3. Conectar CNC shield y motores

Después, conectar la **CNC shield** con los motores nema, el servo motor y los controladores A4988 con sus disipadores de calor. Además de las pilas.

### 4. GRBL Controller

Iniciar el programa controlador GRBL, en nuestro caso hemos utilizado tanto **GRBL controller** como universal gcode sender. Es importante que el puerto coincida con el puerto en el que hemos conectado el arduino y el Baud Rate también.

Port name	COM8	
	Open	Baud Rate 9600 Last State:

Una vez hecho esto, clicar a Open para que se inicie y aparecerán una serie de comandos, si no aparecen hay que introducir el comando \$\$.

```
$0=200.000 (x, step/mm)
$1=200.000 (y, step/mm)
$2=200.000 (z, step/mm)
$3=15 (step pulse, usec)
$4=300.000 (default feed, mm/min)
$5=300.000 (default seek, mm/min)
$6=192 (step port invert mask, int:11000000)
$7=253 (step idle delay, msec)
$8=30.000 (acceleration, mm/sec^2)
$9=0.050 (junction deviation, mm)
$10=0.100 (arc, mm/segment)
$11=25 (n-arc correction, int)
$12=3 (n-decimals, int)
$13=0 (report inches, bool)
$14=1 (auto start, bool)
$15=0 (invert step enable, bool)
$16=0 (hard limits, bool)
$17=0 (homing cycle, bool)
$18=0 (homing dir invert mask, int:00000000)
$19=500.000 (homing feed, mm/min)
$20=500.000 (homing seek, mm/min)
$21=100 (homing debounce, msec)
$22=1.000 (homing pull-off, mm)
```

## 5. Modificar comandos

Las variables que interesantes para este proyecto son del \$0 al \$5 y el \$8

\$0 indica los pasos que da el eje x por mm en nuestro caso lo mas optimo es 300.

\$1 indica los pasos que da el eje y por mm en nuestro caso lo mas optimo es 300 .

(\$0 tiene que ser igual a \$1)

\$2 indica los pasos que da el eje y por mm en nuestro caso lo mas optimo es 10 ya que utilizamos un servo motor para el eje z.

\$3 lo dejaremos igual en 15.

\$4 y \$5 tienen que ser el mismo dato o como mucho \$4 un poco menos en nuestro caso lo mas optimo es 400.

\$8 es la aceleración de nuestros motores, nosotros hemos puesto 100.

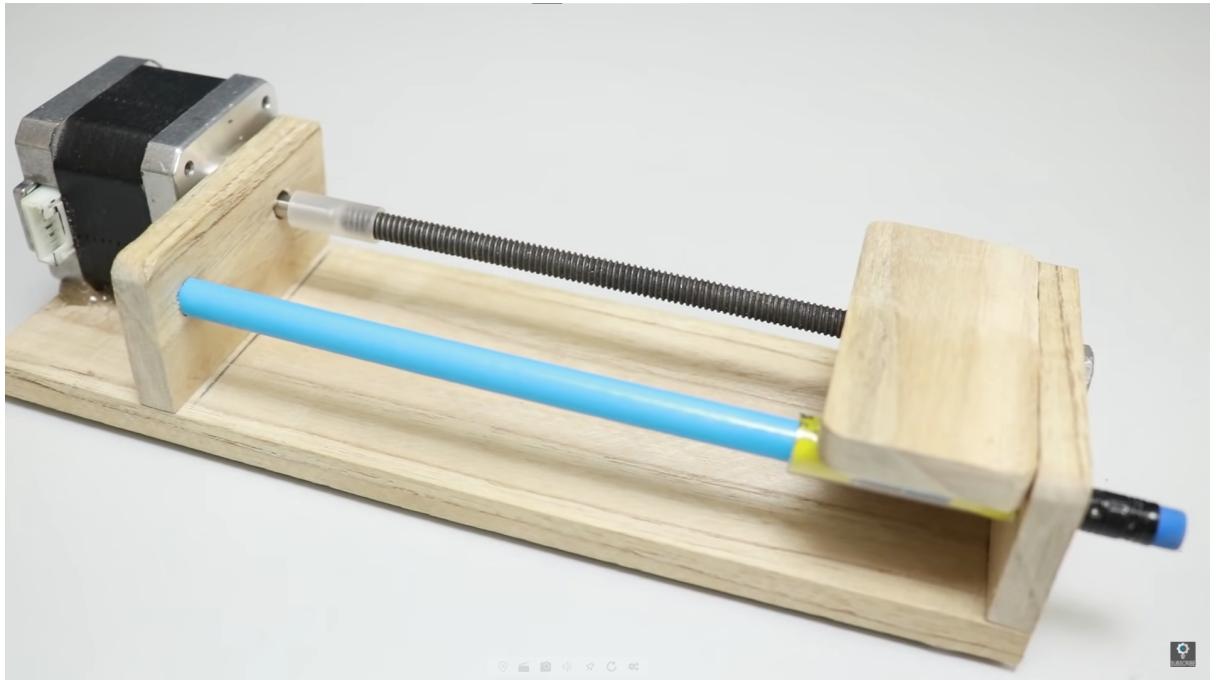
Hay que modificar estos datos para que los motores se muevan lo más rápido posible pero si los sobrecargamos el motor vibrará y perderá velocidad.

## 6. Conectar el bluetooth

Una vez hecho esto conectar el bluetooth al CNC shield. Buscar el bluetooth con el móvil e iniciar la aplicación GRBL controller en android (se puede descargar en google play). Desde aquí se puede mandar la imagen .ng para que comience a dibujar o moverlo solamente en el eje x o el eje y.

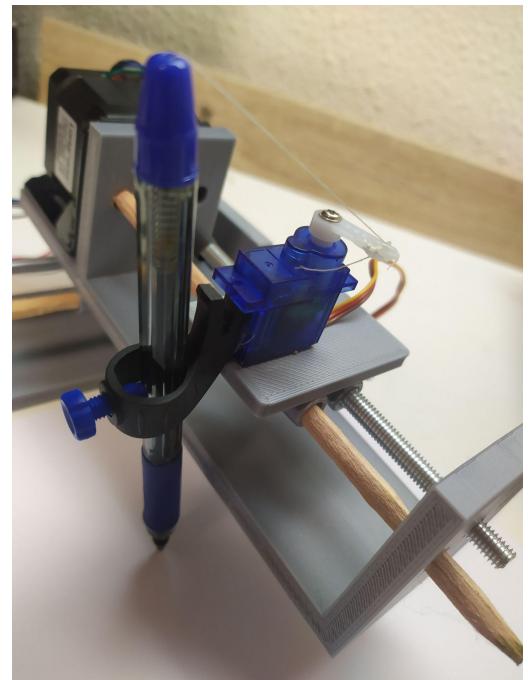
## MECANISMO

Tanto el eje x como el eje funcionan igual, el motor al moverse hace mover un palo con rosca que le hemos pegado, este mueve las dos tuercas de un lado a otro. Pegándole otra pieza a las tuercas, esta se mueve con las tuercas. El lápiz sirve de apoyo para que la pieza se ponga recta.



En el eje z hemos construido un mecanismo pegando un trozo de hilo a un muelle en el interior de un bolígrafo. Entonces cuando el servo motor mueve este hilo, el muelle subirá y la punta del bolígrafo subirá.

De esta forma el bolígrafo se puede mover sin dibujar y colocarse solo en el punto donde comienza el dibujo



## PROBLEMAS

Tuvimos varios problemas durante el proyecto, el primero fue la placa CNC shield, no sabíamos cómo funcionaba, nos daba errores con el programa controlador de arduino. Tuvimos que buscar la solución pero,

no había mucha información sobre esta placa. Al final descubrimos que había que iniciar el controlador en el programa de arduino.

Luego tuvimos problemas menores, como por ejemplo las piezas que las imprimimos en 3D y tenían que ser perfectas para el funcionamiento de la impresora. Los tornillos hubo que cortarlos y unirlos con goma termoretractil y era complicado porque no quedaba recto.

Aprendimos a controlar los motores, son muy potentes pero tenían que ser precisos. La aplicación GRBL Controller era bastante compleja.

Como último problema el bluetooth como a la mayoría de los compañeros de clase nos ha dado muchos errores.

## COSAS A MEJORAR

El principal problema es que las piezas no son lisas ya que las lijamos a lápiz y las piezas hechas con la impresora no están perfectas, entonces a la hora de moverse no hay una línea recta.

Esto se debe a que estuvimos mucho tiempo para que los motores funcionasen con el bluetooth ya que no hay mucha información de como hacerlo en internet y tuvimos que sacar información de gente que hay hecho proyectos parecidos como una Cnc o una impresora 3D

Otro problema sería que por la limitación de tamaño de la impresora 3D con la que hicimos las piezas el proyecto ha salido más pequeño de lo que pensábamos entonces no puede hacer dibujos muy grandes.

## CONCLUSIÓN

En conclusión estamos contentos con nuestro proyecto ya que ha funcionado, y a pesar de tener múltiples problemas hemos conseguido solucionarlos y seguir adelante. Además hemos aprendido un montón sobre arduino, y todas las cosas que puedes hacer con arduino. Ha sido una tarea difícil, pero con esfuerzo y trabajo hemos conseguido que salga adelante. Aprendiendo no solo de arduino si no de motores, diseño de piezas 3D y uso de herramientas.